

Universidad Carlos III de Madrid

Escuela Politécnica Superior



Trabajo de Fin de Estudios

Grado en Ingeniería Informática

**Desarrollo de un videojuego de
realidad mixta**

Autor: Pedro Sanz Herrero

Tutor: Telmo Agustín Zarraonandia Ayo

RESUMEN

En los últimos años, el auge de las nuevas tecnologías, como la realidad aumentada o la realidad virtual, están cambiando el día a día de muchas personas, y también han llegado al punto de revolucionar el mundo de los videojuegos.

Esta ha sido la causa de que se decidiera desarrollar una aplicación que pudiera hacer uso de estas tecnologías para adaptar o crear juegos con unas mecánicas conocidas o intuitivas para los usuarios, de forma que el aprendizaje de los controles de los mismos y del uso de la realidad aumentada fuera lo más fácil posible para ellos.

En este proyecto se va a desarrollar una aplicación consistente en diversos juegos, diseñada para ser utilizada mediante controles intuitivos a través de la realidad aumentada. Se expondrán las decisiones de diseño tomadas a lo largo del desarrollo para cumplir con los objetivos iniciales que se han planteado para esta aplicación y el proceso del desarrollo de la misma. Finalmente, se explicarán las conclusiones extraídas de este proyecto y de las posibilidades del mismo.

Palabras clave: realidad aumentada, realidad mixta, aplicación, videojuego.

ABSTRACT

In the last few years, the rising of new technologies like augmented reality or virtual reality is changing the everyday life of a lot of people, and has even started to revolutionize the world of video games.

This has been the reason why it has been decided to develop an application that can make use of these technologies to adapt or create games with known or intuitive mechanics for the users, such that the learning of the controls and the use of augmented reality was as simple as possible for them.

In this project an application consisting of several games is going to be developed, designed to be used through intuitive controls that make use of augmented reality. The design decisions taken through the development to fulfill the initial objectives of the application and the development process will be exposed in this document. Finally, the conclusions obtained from this project and its possibilities will be further explained.

Keywords: augmented reality, mixed reality, application, video game.

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS	9
ÍNDICE DE TABLAS	12
1. INTRODUCCIÓN	15
1.1 Contexto	15
1.2 Problemas	16
1.3 Objetivos	17
1.4 Herramientas utilizadas	18
1.5 Estructura del documento	21
2. ESTADO DEL ARTE	23
2.1 Tecnología	23
2.1.1 Definición	23
2.1.2 Historia	24
2.1.3 Ventajas	25
2.1.4 Desventajas	26
2.1.5 Ejemplos	27
2.2 Dispositivos	32
2.2.1 Smartphones	32
2.2.2 Gafas	33
2.2.3 Reloj	34
2.2.4 Consola	36
2.3 Librerías	37
2.3.1 Vuforia	37
2.3.2 Metaio	37
2.3.3 Wikitude	37
2.3.4 Comparación	37

3. ANÁLISIS	39
3.1 Elección de juegos	39
3.2 Requisitos del usuario	41
3.2.1 Capacidades generales	41
3.2.2 Restricciones generales	41
3.2.3 Características del usuario	41
3.2.4 Entorno operativo	42
3.2.5 Entorno de seguridad	42
3.2.6 Suposiciones y dependencias	42
3.2.7 Requisitos específicos	43
3.2.7.1 Requisitos de capacidades	43
3.2.7.2 Requisitos de restricciones	47
3.3 Casos de uso	48
3.4 Requisitos del software	54
3.4.1 Requisitos funcionales	55
3.4.2 Requisitos de desempeño	56
3.4.3 Requisitos operacionales	56
3.4.4 Requisitos de recursos	57
3.4.5 Requisitos de seguridad	58
3.4.6 Requisitos no funcionales	58
3.4.7 Matriz de trazabilidad	60
4. DISEÑO	61
4.1 Arquitectura	61
4.2 Descripción de los componentes	64
4.3 Conceptos	71
5. IMPLEMENTACIÓN	74
5.1 Desarrollo de realidad aumentada	74
5.2 Desarrollo de la aplicación	81

6. PRUEBAS	84
7. CONCLUSIONES	86
7.1 Objetivos cumplidos	86
7.2 Futuras líneas de trabajo	87
8. GESTIÓN DEL PROYECTO	88
8.1 Planificación	88
8.2 Método de desarrollo	90
8.3 Presupuesto	91
9. BIBLIOGRAFÍA	93
ANEXO I: MANUAL DE USUARIO	95
ANEXO II: GLOSARIO	102
DEVELOPMENT OF A MIXED-REALITY VIDEO GAME	104
Introduction	104
Objectives	105
Games election	105
Game concepts	108
Tests	111
Conclusions	113

ÍNDICE DE FIGURAS

Figura 1. MacBook Pro Retina	18
Figura 2. Elephone P9000	19
Figura 3. Epson Moverio BT-200	19
Figura 4. Diagrama del continuo de la virtualidad de Milgram	23
Figura 5. Sensorama	24
Figura 6. KARMA	25
Figura 7. Google Translate haciendo uso de la cámara	27
Figura 8. Interfaz de Star Walk	28
Figura 9. Interfaz de Yelp Monocle	29
Figura 10. Reconocimiento facial y posteriores efectos aplicados con Snapchat Lenses	29
Figura 11. Invizimals en PS Vita	30
Figura 12. Uso del GPS y la cámara en Pokémon GO	31
Figura 13. Interacción de personajes creados con distintos marcadores en Play with Mario	31
Figura 14. Google Glass	33
Figura 15. Distintos modelos de smartwatch	35
Figura 16. PS Vita y Nintendo 3DS	36
Figura 17. Escena de juego de Breakout	39
Figura 18. Escena de juego Labyrinth para iOS	40
Figura 19. Caso de uso de selección de juego	49
Figura 20. Caso de uso de manejo de Arkanoid	49
Figura 21. Caso de uso de manejo del juego del laberinto	50
Figura 22. Diagrama de clases del juego Arkanoid	62
Figura 23. Diagrama de clases del juego del laberinto	63
Figura 24. Marcador elegido para su uso con Vuforia	71
Figura 25. Posicionamiento inicial del texto con las opciones	72

Figura 26. Posicionamiento final del texto con las opciones	72
Figura 27. Diseño de la pelota, paredes, suelo y cajas del juego Hierba Rodadora	73
Figura 28. Lápiz virtual sobre marcador	74
Figura 29. Imágenes de números utilizados como marcadores	75
Figura 30. Proyección de una granja de GREP Player sobre un marcador	76
Figura 31. Representación de la gravedad virtual con marcador horizontal	76
Figura 32. Representación de la gravedad virtual contra la real con marcador no horizontal	77
Figura 33. Representación de la gravedad virtual con cámara vertical	78
Figura 34. Representación de la gravedad virtual contra la real con cámara no vertical	78
Figura 35. Representación de la gravedad real y virtual sin importar el ángulo de la cámara ni del marcador de juego	79
Figura 36. Representación de la gravedad virtual dependiendo del ángulo de la cámara	80
Figura 37. Objetos y jerarquía de la escena de selección de juego	81
Figura 38. Objetos y jerarquía de la escena de selección de dificultad	81
Figura 39. Objetos y jerarquía de la escena del juego Arkanoid	82
Figura 40. Objetos y jerarquía de la escena del juego Hierba Rodadora	83
Figura 41. Juego de Arkanoid en nivel fácil y difícil	84
Figura 42. Juego Hierba Rodadora en nivel fácil y difícil	84
Figura 43. Menú de elección de dificultad desde distintos ángulos	85
Figura 44. Imagen de victoria y derrota en el juego Arkanoid	85
Figura 45. Diagrama de Gantt inicial del proyecto	88
Figura 46. Diagrama de Gantt final del proyecto	89
Figura 47. Método de desarrollo en V	90
Figura 48. Menú de selección de juego	95
Figura 49. Modo de selección de dificultad	96
Figura 50. Menú de selección de dificultad	97

Figura 51. Escenario inicial del juego Arkanoid	98
Figura 52. Modo de juego de Arkanoid	99
Figura 53. Escenario inicial del juego Hierba Rodadora	100
Figura 54. Modo de juego de Hierba Rodadora	101

ÍNDICE DE TABLAS

Tabla 1. Comparación de plataformas de realidad aumentada	38
Tabla 2. Requisito de capacidades UC01	43
Tabla 3. Requisito de capacidades UC02	44
Tabla 4. Requisito de capacidades UC03	44
Tabla 5. Requisito de capacidades UC04	44
Tabla 6. Requisito de capacidades UC05	44
Tabla 7. Requisito de capacidades UC06	45
Tabla 8. Requisito de capacidades UC07	45
Tabla 9. Requisito de capacidades UC08	45
Tabla 10. Requisito de capacidades UC09	45
Tabla 11. Requisito de capacidades UC10	46
Tabla 12. Requisito de capacidades UC11	46
Tabla 13. Requisito de capacidades UC12	46
Tabla 14. Requisito de capacidades UC13	46
Tabla 15. Requisito de restricciones UR01	47
Tabla 16. Requisito de restricciones UR02	47
Tabla 17. Requisito de restricciones UR03	47
Tabla 18. Requisito de restricciones UR04	47
Tabla 19. Requisito de restricciones UR05	48
Tabla 20. Requisito de restricciones UR06	48
Tabla 21. Caso de uso de selección de juego	51
Tabla 22. Caso de uso de manejo de Arkanoid	52
Tabla 23. Caso de uso de manejo del juego del laberinto	53
Tabla 24. Requisito funcional SF01	55
Tabla 25. Requisito funcional SF02	55
Tabla 26. Requisito funcional SF03	55

Tabla 27. Requisito funcional SF04	55
Tabla 28. Requisito funcional SF05	56
Tabla 29. Requisito de desempeño SD01	56
Tabla 30. Requisito operacional SO01	56
Tabla 31. Requisito operacional SO02	56
Tabla 32. Requisito operacional SO03	57
Tabla 33. Requisito de recursos SR01	57
Tabla 34. Requisito de recursos SR02	57
Tabla 35. Requisito de recursos SR03	57
Tabla 36. Requisito de seguridad SS01	58
Tabla 37. Requisito no funcional SN01	58
Tabla 38. Requisito no funcional SN02	58
Tabla 39. Requisito no funcional SN03	58
Tabla 40. Requisito no funcional SN04	59
Tabla 41. Requisito no funcional SN05	59
Tabla 42. Requisito no funcional SN06	59
Tabla 43. Matriz de trazabilidad	60
Tabla 44. Componente CA01	64
Tabla 45. Componente CA02	65
Tabla 46. Componente CA03	65
Tabla 47. Componente CA04	66
Tabla 48. Componente CA05	66
Tabla 49. Componente CA06	67
Tabla 50. Componente CL01	67
Tabla 51. Componente CL02	68
Tabla 52. Componente CL03	68
Tabla 53. Componente CL04	69
Tabla 54. Componente CL05	69

Tabla 55. Componente CL06	70
Tabla 56. Componente CL07	70
Tabla 57. Coste de personal	91
Tabla 58. Coste de hardware y software	92

1. INTRODUCCIÓN

1.1 Contexto

En los últimos años, la tecnología ha sufrido uno de los mayores avances en los últimos años: los llamados teléfonos inteligentes o *smartphones*. Estos dispositivos han conseguido llevar conceptos tales como aplicaciones o internet a un público mucho más amplio que el que las conocía a través de otros dispositivos capaces de utilizarlas, como los ordenadores portátiles o las PDAs (*Personal Digital Assistant* o asistente digital personal).

A fecha de enero de 2016, un 80% de la población adulta utiliza ya un *smartphone* [1]. El hecho de que conseguir estos dispositivos cada vez sea más asequible, y el valor que añade que tantas personas lo utilicen para comunicarse, haciendo así casi necesario disponer de uno para poder comunicarse con ellos, hace que esta cifra siga aumentando cada vez más.

Sin embargo, la comunicación no es el único uso del *smartphone*: desde su invención, las llamadas aplicaciones o *apps* han permitido también llevar utilidades y juegos a cualquier parte a la que se pueda llevar el *smartphone*, convirtiéndose así también en una revolución para el mundo de los videojuegos. La popularidad de las *apps* ha permitido que los usuarios puedan pasar el rato jugando en sus pequeños terminales, sin necesidad de comprar un dispositivo más caro y grande dedicado especialmente para este propósito, como pueden ser las videoconsolas o los ordenadores.

Por otro lado, en esta última década también se ha visto el aumento en una tecnología relacionada directamente con los *smartphones*: los llamados dispositivos ponibles o *wearable technology*, entre los que caben destacar los relojes inteligentes (*smartwatch*) y las gafas de realidad aumentada. A pesar de que estas tecnologías no han tenido la acogida de los *smartphones* debido a problemas como limitaciones del *software*, duración de la batería o el precio, sí que siguen creciendo en el número de usuarios y se prevé un aumento mayor en los próximos años [2].

En el caso de las gafas de realidad aumentada, en concreto, el principal problema por el cual aún no se han extendido es el elevado precio de los dispositivos que se encuentran en el mercado, que hace que el usuario medio no se plantee su compra.

Esto no significa, sin embargo, que en un futuro no vayan a tener una utilidad en el día a día de muchas personas, y con ese pensamiento en mente está creado este proyecto, que pretende demostrar que con una tecnología no muy distante de la que podemos encontrar hoy en día se pueden lograr objetivos que se aprovecharán mejor cuando sea de un uso más extendido.

1.2 Problemas

A pesar de que la realidad aumentada es una tecnología con un futuro muy prometedor, no solo para los juegos sino también para enriquecer distintas experiencias del día a día, cuenta con varios inconvenientes que hacen que no reciba tanta atención como otras nuevas tecnologías.

El principal problema de la realidad aumentada actualmente es el precio. A día de hoy, no se encuentran dispositivos a un precio asequible al consumidor, o que por el precio que cuestan ofrezcan al usuario una experiencia que aporte la inmersión y precisión necesaria por ese precio.

Un segundo problema con la realidad aumentada es que, debido a la mayor facilidad de crear juegos de realidad virtual que luego se controlan con un equipamiento especial para dicha tecnología, y a que al usar un equipamiento dedicado la precisión de los movimientos reales es mayor, pocas de las empresas de videojuegos que se están enfocando a las nuevas tecnologías tienen como objetivo desarrollar para realidad aumentada. Esto se puede comprobar si se observan los múltiples futuros lanzamientos de la industria de los videojuegos relacionados con la realidad aumentada, entre los que se incluye *hardware* como cascos de realidad virtual, comparados con los pocos planeados que hagan uso de la realidad aumentada.

El tercer problema es, como se menciona en el punto anterior, la falta de precisión de la realidad aumentada. Al usar sensores que miden datos directamente del mundo real, en lugar de recibir los datos de un sensor externo dedicado como puede ser un mando, los gestos del mundo real pueden ser malinterpretados o no reconocidos por el dispositivo.

Otro problema que se puede encontrar actualmente es la falta de motivación para utilizar la realidad aumentada. Los dispositivos que más se utilizan actualmente son los *smartphones*, y a la mayoría de aplicaciones no les aporta alicientes el uso de la realidad aumentada, dado que podría ser contraproducente para los usuarios. En el caso de las gafas de realidad aumentada, sin embargo, sí que aporta muchas ventajas el uso de esta tecnología para poder utilizarla en conjunto con lo que se ve en el mundo real. Sin embargo, las gafas de realidad aumentada son un producto que actualmente es demasiado caro para el poco uso del que se puede disfrutar, y en el caso de que pretendan dar una experiencia más inmersiva al usuario se vuelven demasiado aparatosas para poder llevarlas con normalidad en lugares públicos, que es su principal objetivo.

Aparte de los problemas relacionados con la parte tecnológica de la realidad aumentada, también podemos encontrarnos con problemas de aceptación de la tecnología por parte de los usuarios y culturales de la realidad aumentada que hace uso del reconocimiento de gestos o imágenes a través de una cámara.

Uno de estos problemas es la diversidad de gestos utilizados para diferentes usos en diferentes culturas y países. Un gesto en apariencia inocente en España puede

resultar ofensivo en otra cultura, y por lo tanto se debe simplificar lo máximo posible las acciones que se puedan realizar.

Por otra parte, el hecho de enfocar con la cámara del dispositivo para poder hacer uso de la realidad aumentada puede dar lugar a situaciones en las que una tercera persona se sienta ofendida por pensar que se está haciendo un uso indebido de su imagen, o invadiendo su privacidad.

También existen problemas que ciertos usuarios pueden experimentar cuando llevan puestos dispositivos de realidad aumentada como las gafas. En este caso, hay algunas personas que pueden sufrir mareos o dolor de ojos o de cabeza debido a la mezcla de la percepción del mundo real con la del mundo virtual.

Por último, también nos encontramos con los problemas relacionados con el desarrollo de una aplicación. Cada dispositivo tiene unas especificaciones distintas, entre las que se incluyen distinto tamaño y resolución de la pantalla, distinto procesador, una mayor o menor calidad de la cámara... Todos estos factores hacen que sea difícil desarrollar una aplicación que funcione en todos los dispositivos con la misma precisión y fluidez.

1.3 Objetivos

Los principales objetivos de este proyecto son los siguientes:

- **Desarrollo de un juego de realidad aumentada para gafas:** el objetivo principal de este proyecto es el desarrollo de un juego de realidad mixta, más concretamente de realidad aumentada, que proporcione una mayor inmersión al usuario que la que puede proporcionar la pequeña pantalla de un *smartphone*. Al crearla para gafas de realidad aumentada se puede conseguir esto, pero también hace necesario que se pueda manejar completamente sin depender de ningún elemento virtual. Esto significa que todos los controles ocurren a través de movimientos típicos del mundo real, como tocar o inclinar, en lugar de tocar una pantalla ya que las manos del usuario están libres para una mayor inmersión.
- **Mínima interferencia con el mundo real:** otro de los objetivos es que dicha aplicación intente interferir lo menos posible con otras actividades que puedan estar realizándose a la vez que el juego esté en marcha, o que puedan surgir mientras el juego esté en marcha. Dado que la aplicación está pensada para utilizarse en cualquier momento y lugar, si requiriera una equipación adicional al dispositivo no cumpliría con este objetivo.
- **Portabilidad a *smartphones*:** el tercer y último objetivo es que, a pesar de ser desarrollada para unas gafas de realidad aumentada principalmente, pueda ser usada también con un *smartphone*. La razón de este objetivo es que, aunque utilizar el juego con un *smartphone* en la mano reduce la jugabilidad de la aplicación (y si se utilizan unas gafas de realidad aumentada se disponen de más elementos en el mundo real para tener un mejor control y un mayor campo de visión), una gran cantidad de la población utiliza los

smartphones diariamente. A fin de hacerlo disponible a un público mayor y para poder tomar imágenes más claras de situaciones del juego que, de otra forma, no podrían verse igual debido a que la imagen debe duplicarse en unas gafas de realidad aumentada para producir sensación de profundidad, la aplicación se desarrollará también para *smartphones*.

1.4 Herramientas utilizadas

Las herramientas utilizadas para hacer posible el desarrollo de este proyecto han sido las siguientes:

- **Ordenador portátil:** todo el desarrollo de la aplicación, así como gran parte de la redacción de este informe, se ha realizado desde este ordenador. En concreto se trata de un Macbook Pro Retina de principios de 2015, con un procesador Intel Core i5 de 2,7 GHz, una memoria de 8 GB de 1867 MHz DDR3 y una tarjeta gráfica Intel Iris Graphics 6100 de 1536 MB, representado en la figura 1. El sistema operativo de dicho ordenador es OS X El Capitan, versión 10.11.6.



Figura 1. MacBook Pro Retina

- **Teléfono móvil:** las pruebas de la aplicación en un *smartphone* han sido realizadas para comprobar de forma más fácil algunas de las funcionalidades y para obtener imágenes más claras del funcionamiento de ésta. Todas se han realizado en un Elephone P9000 con un procesador Helio P10 (MTK6755) Octa Core, una GPU (*Graphical Processing Unit* o unidad de procesamiento gráfico) ARM Mali-T860 MP2 de 700 MHz y una memoria RAM (*Random-Access Memory* o memoria de acceso aleatorio) de 4 GB,

representado en la figura 2. El sistema operativo utilizado para realizar las pruebas es Android 6.0 *Marshmallow*.



Figura 2. Elephone P9000

- **Gafas de realidad aumentada:** el dispositivo para el que la aplicación ha sido principalmente diseñada y en el que se ha probado después de completar el desarrollo es un Epson Moverio BT-200, representado en la figura 3. El sistema operativo que tenían instalado estas gafas es Android 4.0 *Ice Cream Sandwich*.



Figura 3. Epson Moverio BT-200

- **Plataforma de desarrollo:** para poder desarrollar la aplicación, y también como motor de juego, se ha utilizado la plataforma de desarrollo Unity. Debido a problemas de soporte de Vuforia con las versiones más recientes

de Unity, el juego se creó con la versión Unity 5.3.1p3 para el sistema operativo OS X.

- **Plataforma de realidad aumentada:** debido a la complejidad del reconocimiento de imágenes para poder incorporar la realidad aumentada al juego, el uso de una plataforma que permita desarrollar de forma intuitiva aplicaciones de realidad aumentada fue necesario. Para este proyecto la plataforma elegida fue Vuforia, y en concreto se utilizó la versión Vuforia 5.5.9 para Unity.
- **Kit de desarrollo de software:** para poder convertir la aplicación del proyecto en Unity a un juego que pueda ser jugado en Android, es necesario utilizar el SDK (*Software Development Kit* o kit de desarrollo de *software*) de Android. La mínima versión requerida de la API (*Application Programming Interface* o interfaz de programación de aplicaciones) de Android para que el juego pueda funcionar es Android 2.3.1 *Gingerbread*, que corresponde con la versión 9 de la API.
- **Procesador de texto:** para organizar y escribir este informe han sido necesarios dos procesadores de texto distintos.

El primero de ellos es Google Docs, un procesador de texto de la compañía Google basado en el almacenamiento en la nube de los documentos así como en la sincronización entre colaboradores editando a la vez un mismo documento. Todo la escritura del documento se ha realizado a través de este procesador, debido a la facilidad que proporciona poder escribir desde cualquier dispositivo conectado a la red y a la capa de seguridad que proporciona el estar almacenado en un disco personal en la nube en lugar de en un archivo local de un dispositivo, que tiene mayor predisposición a sufrir algún problema.

El segundo de los procesadores utilizados es Microsoft Word, propiedad de la compañía Microsoft, que ha sido utilizado para hacer retoques menores en algunos aspectos del documento, debido a que posee funcionalidades no disponibles en Google Docs. La versión utilizada de este producto es Microsoft Word 365 para Mac, versión 15.25.

- **Copia de seguridad:** como respaldo tanto para el desarrollo del proyecto en sí como para mantener seguro el informe del proyecto, se utilizó la plataforma online Google Drive, de la compañía Google. Para mantener todo el proyecto sincronizado automáticamente y así aprovechar mejor la funcionalidad de esta plataforma se usó el programa externo Expandrive, que permite mantener archivos locales sincronizados con varios servidores externos, entre los que se incluye Google Drive.
- **Organizador de bibliografía:** para mantener un formato adecuado de la bibliografía de este documento y tener todos los documentos consultados disponibles de forma rápida se ha usado el programa Mendeley, que además permite mantener sincronizados estos datos entre varios dispositivos. La

versión de escritorio utilizada ha sido Mendeley Desktop para Mac, versión 1.16.1.

1.5 Estructura del documento

En este punto se va a proceder a describir cómo está organizado este documento, incluyendo una breve descripción de los temas que se van a cubrir en cada capítulo para hacerse una idea general de la estructura tanto de la aplicación como de la memoria:

- **Introducción:** este capítulo incluye una descripción de las motivaciones que han llevado a desarrollar esta aplicación en concreto, así como un listado de problemas relacionados con varios aspectos que pueden llegar a afectar de alguna forma al funcionamiento y acogida de la aplicación. Asimismo, se exponen los objetivos que se tuvieron en mente cuando se decidió empezar a crear el juego y los elementos, tanto de *hardware* como de *software*, utilizados para hacer posible que la aplicación y esta memoria existan. Finalmente, también se expone un resumen de la estructura del documento con breves descripciones de los contenidos de cada capítulo.
- **Estado del arte:** en este capítulo se explicarán diversas tecnologías que actualmente están haciendo uso de la realidad aumentada, haciendo hincapié en cuáles son las más utilizadas y qué aspectos del mundo real utilizan. También se describirán diversas herramientas que están siendo utilizadas hoy en día con el propósito de facilitar el desarrollo de aplicaciones que hagan uso de la realidad aumentada, y se explicará qué motivos han llevado a elegir una de estas tecnologías en concreto.
- **Análisis:** esta sección incluye una revisión de los requisitos de los usuarios que se plantearon antes de desarrollar la aplicación, así como diversos requisitos del sistema que se definieron con el objetivo de que la aplicación los cumpliera a partir de los requisitos de usuario. Esta sección incluye una matriz de trazabilidad que relaciona ambos tipos de requisitos, y casos de uso derivados de los requisitos.
- **Diseño:** una vez analizados los requisitos de software en el capítulo anterior se procede a diseñar la arquitectura y el diseño detallado de la aplicación, con ejemplos que incluyen diagramas de clase, la descripción de los componentes que lo conforman y decisiones de diseño tomadas para la aplicación.
- **Implementación:** este capítulo incluye los proyectos previos que se realizaron para comprobar las posibilidades del desarrollo con la tecnología de la realidad aumentada, así como información de cómo sirvieron esos proyectos previos para añadir funcionalidades concretas de la plataforma de desarrollo de realidad aumentada Vuforia al proyecto.

- **Pruebas:** esta sección contempla diversas pruebas que se han realizado a la aplicación una vez finalizada, para comprobar que cumple con los objetivos planteados y que es un producto terminado.
- **Conclusiones:** un resumen de los objetivos iniciales que se han llegado a cumplir, los requisitos que han conseguido verse satisfechos y posibles mejoras que puedan implementarse en el futuro para mejorar la aplicación.
- **Gestión del proyecto:** en este capítulo se mostrará una estimación del esfuerzo realizado a lo largo del desarrollo del proyecto, así como una proyección de los costes que ha supuesto su desarrollo.
- **Bibliografía:** una lista de todas las referencias bibliográficas empleadas a lo largo del documento, con enlaces a sus respectivas fuentes en caso de estar disponibles en internet.
- **Anexos:** esta sección final consta de dos apéndices. El primero de ellos consiste en un manual de usuario, que contiene información sobre la instalación y uso del juego. El segundo anexo incluye y define los términos más relevantes usados a lo largo de la memoria.
- **Development of a mixed-reality video game:** pequeña versión en inglés de este informe, con los aspectos más importantes como un resumen, los objetivos principales, los resultados y las conclusiones.

2. ESTADO DEL ARTE

2.1 Tecnología

Para poder explicar por qué se han elegido los dispositivos y librerías seleccionados para poder desarrollar la aplicación que hace uso de la realidad aumentada, primero debemos ver de qué se trata esta tecnología y cuándo se empezó a utilizar; las ventajas que puede aportar el uso de esta y algunos ejemplos prominentes para hacernos una idea de lo avanzado y extendida que está. En primer lugar, empezamos respondiendo a la pregunta “¿Qué es la realidad aumentada?”.

2.1.1 Definición

Para poder entender el concepto de realidad aumentada, primero se deben definir el mundo real y el mundo virtual.

El mundo real es aquel que percibimos en el que los objetos tienen una calidad practicable y verificable, y que no está basado en lo imaginario, teórico o no físico [3]. Por otro lado, el mundo virtual es un mundo generado por ordenador, en el que las leyes físicas están programadas y cada persona aparece representada como un avatar [4].

Entre estos dos mundos, sin embargo, existe un espectro en el que se entremezclan en proporciones distintas. Todo este espectro es conocido como realidad mixta, como se ilustra en la figura 4.



Figura 4. Diagrama del continuo de la virtualidad de Milgram [5]

Dentro de la realidad mixta, y dependiendo del grado en el que se utilicen los elementos virtuales y reales, se pueden encontrar la realidad aumentada y la virtualidad aumentada. De estos dos conceptos, el de realidad aumentada es el que va a ser explorado.

La realidad aumentada consiste en “aumentar” o suplementar elementos físicos del mundo real, vistos directamente o indirectamente, con elementos del mundo virtual, tales como vídeo, sonido, gráficos o datos de localización de GPS (*Global Positioning System* o sistema de posicionamiento global) [6, 7].

2.1.2 Historia

El primer invento más parecido a la realidad aumentada que se creó fue el llamado Sensorama, una máquina de un aspecto parecido a una recreativa *arcade* en la que el usuario se sentaba e introducía la cabeza para ver un metraje de un camino en bicicleta en primera persona. Esta máquina emulaba vibraciones y sensaciones como el olor o el viento, y por lo tanto se acerca más a la realidad virtual debido a que sustituye casi todas las sensaciones del usuario por otras virtuales en lugar de ampliar una experiencia real. El prototipo del Sensorama fue creado en 1962 por Morton Heilig [8], y se puede apreciar en la figura 5.



Figura 5. Sensorama

No fue hasta mucho más adelante, en 1990, cuando Tom Caudell acuñó el término de realidad aumentada para describir un monitor digital usado por electricistas de aeronaves que mezclaba gráficos virtuales en el mundo real. Dichas gafas

mostraban instrucciones de cómo realizar determinadas, superpuestas sobre la imagen real de la maquinaria sobre la que tenían que operar.

Suponiendo que dichas gafas se correspondieran con el primer ejemplo de realidad aumentada, también surgió otro invento en la misma época que utilizaba un mecanismo similar: un dispositivo HMD (*Head-Mounted Display* o, simplemente, gafas de realidad virtual) que daba instrucciones en formato de imágenes sobre cómo recargar una impresora 3D. Este invento de la Universidad de Columbia fue llamado KARMA (*Knowledge-based Augmented Reality for Maintenance Assistance* o realidad aumentada basada en el conocimiento para la asistencia en el mantenimiento) [8], que se muestra en la figura 6.



Figura 6. KARMA [9]

En la actualidad, la realidad aumentada se usa profesionalmente sobre todo en el ámbito de la aeronáutica, con el formato de HUD (*Head-Up Display* o visualización cabeza-arriba) para aportar información extra al piloto sin que este tenga que apartar la vista [10].

2.1.3 Ventajas

Algunas de las ventajas del uso de la realidad aumentada frente a otras tecnologías son:

- **Movilidad:** en comparación con la realidad virtual, que requiere que dejes de ser consciente de lo que te rodea, en la realidad aumentada el mundo que predomina es el real. Esto permite que se pueda utilizar en cualquier parte, dado que el usuario siempre tiene que ser capaz de percibir el mundo real para poder disfrutar de ella [11].
- **Información:** el principal propósito de la realidad aumentada es enriquecer el mundo real. La cantidad de información disponible en el mundo virtual es infinita, y el hecho de poder acceder a la información que el usuario quiera obtener en ese momento sin perder información del mundo real a cambio no hace sino aumentar la experiencia.

El ejemplo más importante de lo crucial que puede ser la información extra es el de la ARAS (*Augmented Reality-Assisted Surgery* o cirugía asistida por realidad aumentada), en la que un cirujano utiliza unas gafas de realidad aumentada para obtener información adicional sobre el estado del paciente por medio de imágenes virtuales superpuestas sobre la imagen real del paciente.

- **Comunicación:** poder acceder a parte del mundo virtual no solo proporciona información al usuario, también aumenta sus posibilidades de comunicación y de interacción social.
- **Facilidad:** la realidad aumentada permite realizar tareas que el usuario no tiene por qué saber realizar de otra forma. Cada vez existen más y más aplicaciones que contienen instrucciones sencillas sobre cómo realizar ciertas tareas, mostradas de una forma intuitiva superpuestas sobre el mundo real para poder seguir instrucciones fácilmente [12].
- **Experiencia personal:** la realidad aumentada está pensada para poder mostrar la información que más convenga al usuario. La personalización de esta información hace que cada persona pueda tener una experiencia diferente y única según sus preferencias [13].
- **Entretenimiento:** no todas las ventajas de la realidad aumentada están relacionadas con la información o la comunicación. El ocio es también una parte fundamental de la vida del ser humano, y como tal la realidad aumentada permite añadir entretenimiento a situaciones del mundo real del que de otra forma no dispondrían.

2.1.4 Desventajas

El uso de la realidad aumentada también tiene sus desventajas, entre las que se encuentran los siguientes:

- **Percepción real:** a pesar de que la idea de la realidad aumentada es que proporcione información adicional sin ser intrusiva con el mundo real, fijarse en esta información adicional puede suponer distracciones de elementos del mundo real.
- **Difusión:** a pesar de ser una tecnología en aumento, aún no está suficientemente extendida como para que las aplicaciones se desarrollen de forma común para utilizar la realidad aumentada.
- **Coste:** la pequeña difusión de la realidad aumentada con respecto al alcance de otras tecnologías, como los videojuegos o los *smartphones*, hacen que el desarrollo de aplicaciones y juegos que hagan uso de la realidad aumentada sea más costoso que el desarrollo de aplicaciones que no la utilicen.
- **Procesamiento:** la realidad aumentada tiene una curva de aprendizaje que puede hacer que sea más complicada de utilizar para personas no

familiarizadas con un alto nivel de tecnología, pudiendo llegar a hacer sentir incómodo al usuario.

- **Novedad:** la realidad aumentada es una tecnología relativamente nueva y en desarrollo. En caso de querer obtener información del mundo real, la precisión puede ser menor comparada con la de otras tecnologías que utilicen receptores especializados, como sensores [14].

2.1.5 Ejemplos

Para poder hacernos una idea de lo extendida que se encuentra la realidad aumentada en la sociedad de hoy en día, es necesario que sepamos qué aplicaciones existentes son las más extendidas que usan esta tecnología.

Entre los ejemplos de aplicaciones más importantes o prominentes de la realidad aumentada que se usan comúnmente se encuentran las siguientes:

- **Google Translate:** la compañía Google adquirió en 2014 la compañía Quest Visual, propietaria de la aplicación Word Lens, para incorporar su tecnología a la aplicación Google Translate. Esta aplicación permite traducir en tiempo real todo texto que la cámara del dispositivo detecte, mostrando la misma imagen que la que la cámara capta en la pantalla del dispositivo pero con todo el texto traducido, como se ve en la figura 7. Esta aplicación hace uso de la cámara para permitir la realidad aumentada. Está disponible para *smartphones*, *tablets* y Google Glass.

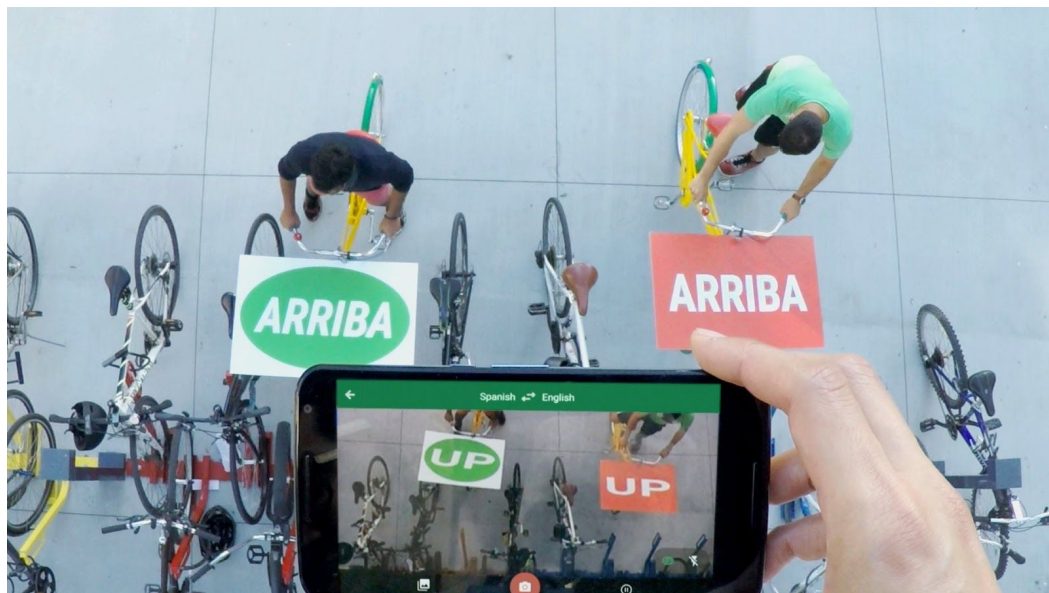


Figura 7. Google Translate haciendo uso de la cámara

- **Star Walk:** diseñada para que tanto expertos como gente que no sabe de astronomía puedan utilizarla, Star Walk permite mostrar de forma fácil las constelaciones y cuerpos celestes que se encuentran a la vista del usuario. Para lograr esto, utiliza una combinación de datos recibidos de la cámara del dispositivo, el giroscopio y el GPS para obtener la localización y orientación del dispositivo. Sobre la imagen de la cámara superpone un mapa virtual del

cielo que permite al usuario distinguir en tiempo real los cuerpos celestes. Un ejemplo del mapa celeste de la aplicación se encuentra en la figura 8. Está disponible para *smartphones* y *tablets*.

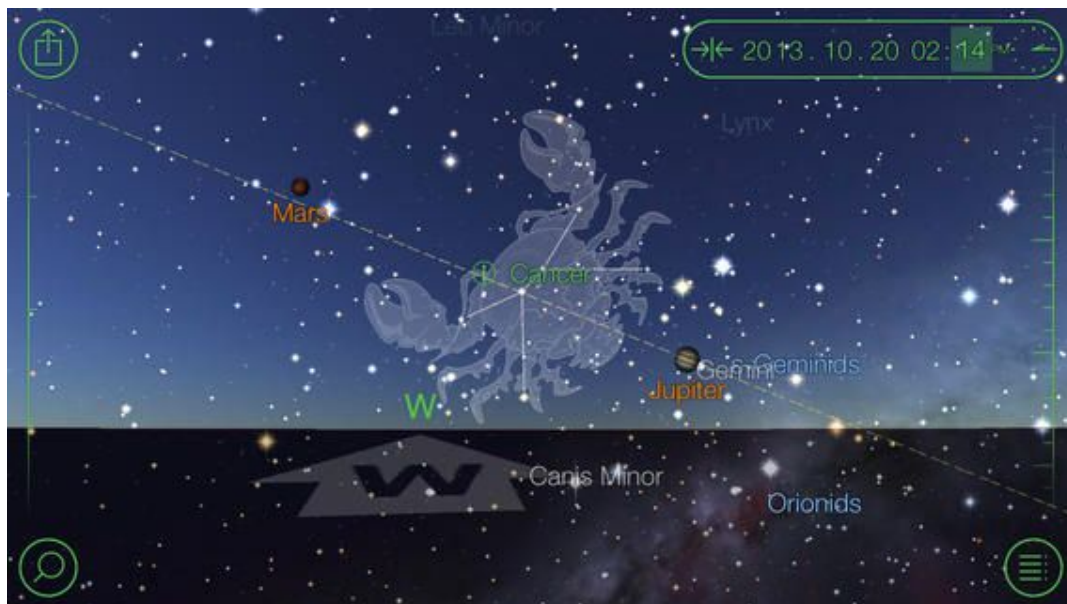


Figura 8. Interfaz de Star Walk

- **Yelp Monocle:** la famosa aplicación de reseñas de restaurantes y otros tipos de negocios Yelp tiene una función poco usada que permite que los usuarios puedan usar la realidad aumentada para orientarse con mayor facilidad. Yelp Monocle permite obtener datos del GPS y el giroscopio del dispositivo para aproximar la localización y orientación del usuario, y poniendo carteles que apuntan a locales cercanos sobre la imagen que recibe de la cámara hace que sea más fácil orientarse, como si se trataran de carteles puestos en la calle, tal y como se puede ver en la figura 9. Aunque Yelp está disponible en multitud de dispositivos, la funcionalidad Monocle solo se encuentra en *smartphones* y *tablets*.

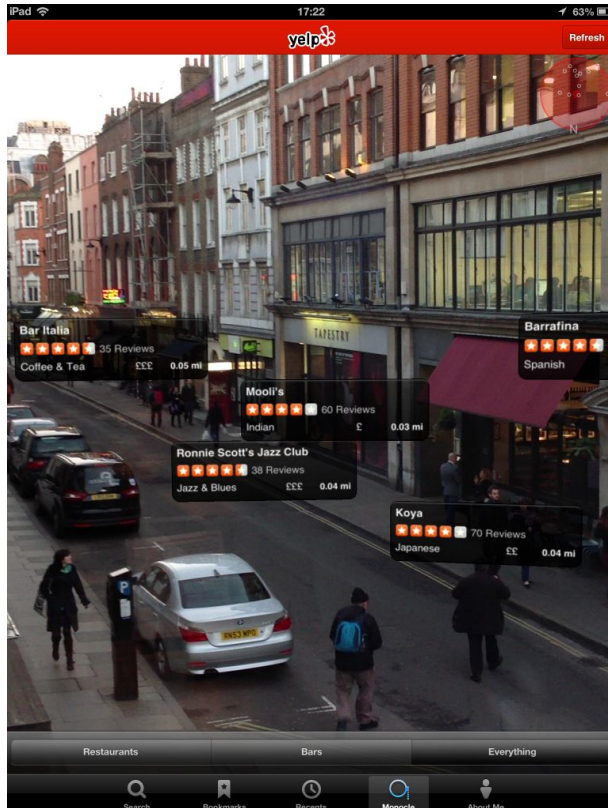


Figura 9. Interfaz de Yelp Monocle

- **Snapchat Lenses:** un ejemplo más común es la función Snapchat Lenses de la aplicación Snapchat. Esta función permite obtener una imagen de la cara y aplicar sobre ella una serie de efectos como máscaras o disfraces, representado en la figura 10. Al aplicar los efectos sobre un avatar creado a partir de la imagen de la cámara, el mundo virtual predomina y por lo tanto se aleja ligeramente de la realidad aumentada y se inclina más hacia la virtualidad aumentada. Está disponible para *smartphones* y *tablets*.

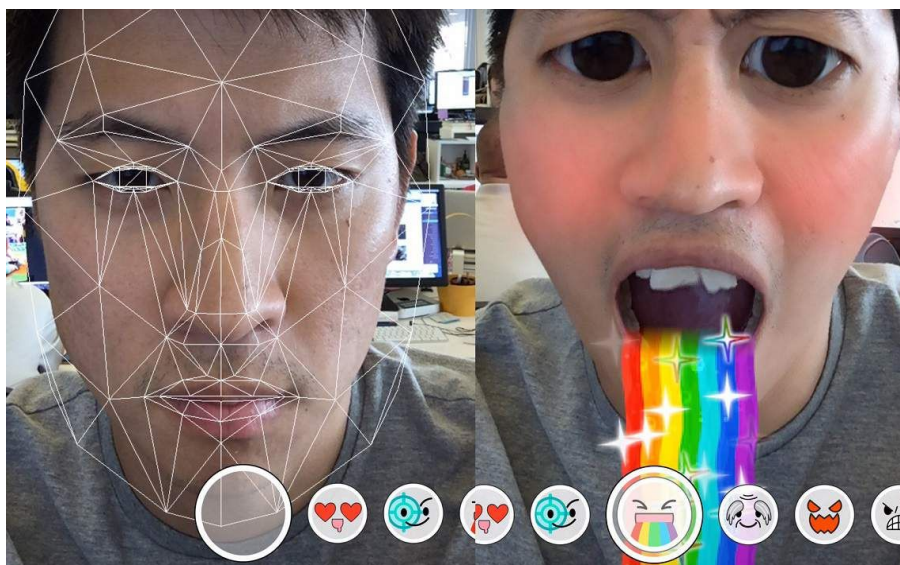


Figura 10. Reconocimiento facial (izquierda) y posteriores efectos aplicados (derecha) con Snapchat Lenses

- **Invizimals:** esta exitosa saga de videojuegos utiliza la cámara para permitir encontrar criaturas ocultas, llamadas Invizimals, en el mundo real. La información se obtiene a través de una cámara, pudiendo usar también marcadores especiales para que la cámara reconozca y actúe en consecuencia. Está disponible para PSP (*PlayStation Portable*) y generaciones posteriores, como se muestra en la figura 11.



Figura 11. Invizimals en PS Vita

- **Pokémon GO:** la aplicación más nueva y de más éxito de las descritas, ha sido creada en 2016 por Niantic después de su último juego de realidad aumentada, Ingress.

El concepto del juego consiste en capturar unas criaturas llamadas Pokémon en el mundo real, a base de rastrearlas mediante el GPS del dispositivo y combatir con ellas a través de la cámara del dispositivo, usando el giroscopio del mismo para permitir dibujar una imagen estable del Pokémon sobre la recibida por la cámara (figura 12). Esta aplicación está disponible para *smartphones*.



Figura 12. Uso del GPS (izquierda) y la cámara (derecha) en Pokémon GO

- **Photos with Mario:** esta aplicación permite sacar fotos del mundo real que incluyen personajes del mundo de Mario Bros. en las fotografías, dibujados digitalmente gracias a unos marcadores que se colocan en el mundo real donde se quiera que el personaje aparezca, tal y como muestra la figura 13. Esta aplicación está disponible para Nintendo 3DS.



Figura 13. Interacción de personajes creados con distintos marcadores en Play with Mario

2.2 Dispositivos

A continuación se van a comparar las diferentes características que cumplen diversos dispositivos que pueden usarse para una aplicación de realidad aumentada, explicando cuáles de ellas son útiles para el propósito de este proyecto y por qué se eligieron los que finalmente han sido utilizados.

2.2.1 Smartphones

El dispositivo electrónico más extendido, ha pasado a formar parte de la vida diaria de la mayoría de las personas. La variedad de *smartphones* disponibles en el mercado hace que sea muy difícil describir las características generales, pero todos ellos tienen algunas en común.

Los *smartphones* están dotados de un sistema operativo. Los más extendidos actualmente son Android, desarrollado por la compañía Google, e iOS, desarrollado por Apple. La gran mayoría de los *smartphones* vendidos actualmente tienen el sistema operativo Android, siendo un 86.2% de los vendidos en el segundo cuarto de 2016 teléfonos Android. Los iPhone, dotados de iOS, fueron el 12.9% de las ventas de *smartphones* [15].

Existen otros *smartphones* con sistemas operativos diferentes, como los Windows Phone, pero con un porcentaje de ventas en el último cuarto de menos del 1%. Por este motivo, solo se tendrán en cuenta los que dispongan de Android o iOS.

Por lo general, los *smartphones* disponen de una gran cantidad de sensores que permiten el desarrollo de aplicaciones de realidad aumentada: giroscopio, acelerómetro, brújula, GPS, micrófono, cámara... Todas estas posibilidades hacen que sean una gran opción para crear este tipo de aplicaciones.

Por otro lado, la extensión del uso de estos dispositivos permiten también que el uso de las aplicaciones que se utilizan en estos se extienda de manera rápida. Al estar tan extendidos también se hace más fácil el desarrollo. Gracias al SDK (*Software Development Kit* o kit de desarrollo de *software*) de Android y de iOS, así como a la multitud de información sobre estos que se puede encontrar *on-line*, la creación de una aplicación para estos sistemas operativos se hace de forma más cómoda, incluso con una aplicación externa que utilice este SDK como Unity.

La mayor desventaja de los *smartphones* respecto a la realidad aumentada es el hecho de tener que sujetarlos con una mano. Al estar diseñados para ser usados con una mano, las posibilidades de usarlos para una aplicación que deje ambas libres para interactuar con elementos virtuales en el mundo real son escasas.

Así mismo, al estar diseñados para mostrar en su pantalla los elementos, se hace difícil también la inmersión del usuario, dado que los detalles del mundo real que debe percibir para añadir información virtual los recibe a través de la pantalla, y no del mundo que le rodea.

A pesar de estas desventajas, el motivo por el que se decidió que la aplicación se desarrollara para *smartphones* Android es debido a la gran cantidad de

documentación disponible, así como a la facilidad de hacer pruebas para detectar posibles fallos en la aplicación, y por la cantidad de datos que se pueden obtener del mundo real a través de un teléfono.

2.2.2 Gafas

Las gafas de realidad aumentada actualmente son un bien no común entre el público. El mercado actualmente está más centrado en la realidad virtual, con productos como Oculus Rift, PlayStation VR, Gear VR o HTC Vive. Incluso Google comercializó las Google Cardboard, que permiten que por un precio inferior a 20€ cualquier *smartphone* pueda convertirse en unas gafas de realidad virtual con las aplicaciones adecuadas.

Sin embargo, también ha habido avances importantes en el campo de la realidad aumentada. El ejemplo más conocido son las llamadas Google Glass, unas gafas de realidad aumentada que disponen de un pequeño ordenador en el lado derecho. En la lente derecha también se sitúa una pequeña pantalla para una experiencia de realidad ininterrumpida que no ocupa todo el campo de visión, sino una pequeña porción. Esto se puede observar en la figura 14.



Figura 14. Google Glass

Otros ejemplos de gafas de realidad aumentada que también han llamado la atención son las Epson Moverio BT-200, en las cuales el procesamiento se hace a través de un dispositivo de apariencia similar a un *smartphone* conectado a las gafas a través de un cable. En el caso de estas gafas, la realidad aumentada sí puede ocupar todo el campo de visión debido a que una pantalla transparente se coloca junto con las lentes.

En el caso de las Epson Moverio, el sistema operativo con el que actúan es Android 4.0 *Ice Cream Sandwich*, mientras que las gafas Google Glass disponen de una versión modificada de Android para mejorar el funcionamiento llamada Glass OS. Debido a la similitud de funcionamiento con un *smartphone*, las gafas disponen de todos los sensores con los que cuenta un *smartphone*, entre ellos cámara, GPS, brújula, giroscopio, acelerómetro, micrófono... Esto hace que el desarrollo para estas plataformas sea parecido al de Android, con la excepción de que en el caso

de las Epson Moverio la imagen virtual proyectada debe ser modificada ligeramente debido al modo de visión de estas gafas.

Las Google Glass disponen de una única pantalla situada en la esquina superior derecha de la lente derecha, proporcionando así información adicional del mundo real pero sin posibilidades de superponer dicha imagen con la que se percibe del mundo real a través de las gafas. Es por esto que pueden incluso no ser consideradas unas gafas de realidad aumentada, ya que los elementos virtuales y los reales no coexisten en un mismo espacio sino que se ven a través de una pequeña ventana.

Las Epson Moverio, por otro lado, cuentan con una pantalla superpuesta en cada lente. En cada una de esas pantallas se muestra la mitad correspondiente de lo que sería la imagen mostrada en un dispositivo Android común, para que con la configuración adecuada estas imágenes ayuden a formar una misma imagen tridimensional en la mente del usuario.

El modo de control general de ambos dispositivos es parecido: en el caso de Google Glass, una pequeña pantalla táctil situada en la patilla derecha de las gafas permite realizar unos controles básicos. Las Epson Moverio requieren tener el dispositivo conectado por un cable a las gafas en la mano, ya que es éste el que dispone de la pantalla táctil. Sin embargo, dado que una vez iniciada la aplicación el control a través de los controles táctiles tradicionales de Android es innecesario, el hecho de que el dispositivo requiera una mano para ser usado es irrelevante.

Debido a que el propósito de este proyecto es el desarrollo de una aplicación inmersiva, y esto requiere que el usuario pueda ver el contenido virtual como si se encontrara en el mundo real, las gafas elegidas para la realización del proyecto han sido las Epson Moverio BT-200.

2.2.3 Reloj

Los *smartwatches* (figura 15) son unos dispositivos con una funcionalidad a un *smartphone*, con la diferencia de que tienen la forma de un reloj de pulsera, y normalmente cuentan con unas capacidades reducidas con respecto a los *smartphones* debido a su menor tamaño y componentes menos potentes.



Figura 15. Distintos modelos de smartwatch

Al contrario que los *smartphone*, una gran cantidad de sistemas operativos diferentes para estos dispositivos se encuentran en el mercado. Entre ellos siguen estando Android Wear y Watch OS, subconjuntos de Android e iOS respectivamente, pero también se pueden encontrar otros como Pebble OS, WebOS o *Microsoft's Wearable Architecture* (que permite ejecutar aplicaciones si está emparejado con un *smartphone*).

Los *smartwatches* están cobrando popularidad en los últimos años, lo cual ha sido una de las causas para considerarlos en este proyecto. Sin embargo, la falta de funcionalidad así como la dificultad para aplicar tecnología de realidad aumentada a estos dispositivos acabó por descartarlos de la elección.

A pesar de que cuentan con casi todos los sensores de los *smartphones* (entre ellos GPS, acelerómetro y brújula), también cuentan con sus desventajas, entre las que se encuentran que una de las manos debe encontrarse en reposo para poder observar la pantalla. También es necesario observar la pantalla para poder percibir el mundo virtual, lo cual impide concentrarse en el mundo real que se puede observar.

Por último, algunos de los *smartwatch* que se encuentran en el mercado requieren estar emparejados a un *smartphone* para tener todas las funcionalidades disponibles. Este hecho, dado que la idea del proyecto es depender del menor número de dispositivos posible para disfrutar de la experiencia, también fue decisivo a la hora de decidir no desarrollar la aplicación para *smartwatch*.

2.2.4 Consola

La primera consola de sobremesa en introducir la realidad mixta de forma exitosa en el mundo de los videojuegos fue la PlayStation 2 con el videojuego EyeToy: Play en 2004. Este juego requería de la cámara externa EyeToy para poder controlar un avatar creado en la consola que podía jugar a diversos juegos. Después del éxito de ésta, se sucedieron diversos periféricos que permitían al usuario controlar con movimientos a sus personajes en las consolas, como es el caso de la Wii de Nintendo, Kinect de Microsoft o PlayStation Eye.

Mientras las consolas de sobremesa se centraron en la virtualidad aumentada, las consolas portátiles incorporaron poco a poco elementos de realidad aumentada en los videojuegos. Este es el caso de la Nintendo DSi, que continuó con la Nintendo 3DS, o la PSP (a través de una cámara externa), que continuó con la PS Vita (*PlayStation Vita*). Estas consolas (figura 16), a través de sus cámaras, desarrollaron juegos que hacían uso de ellas para introducir elementos virtuales en el mundo real.



Figura 16. PS Vita (izquierda) y Nintendo 3DS (derecha)

Sin embargo, el hecho de que se necesiten ambas manos para controlar la consola, así como no disponer de tanta variedad de sensores (sin acelerómetro ni giroscopio) y la dificultad del desarrollo para estas plataformas hizo que fueran descartadas de la selección de dispositivos en los que desarrollar el proyecto.

2.3 Librerías

Por último, una vez elegido el dispositivo en el que se va a desarrollar el proyecto, era necesario elegir en qué SDK nos vamos a apoyar para la detección de datos del mundo real. Actualmente existen multitud de tecnologías que se están desarrollando para esto, pero de ellas hay tres importantes que serán analizadas a continuación:

2.3.1 Vuforia

Anteriormente conocida como QCAR (*Qualcomm Augmented Reality*), esta plataforma de desarrollo de aplicaciones fue creada en 2011 por la empresa Qualcomm, y adquirida en 2015 por la compañía PTC (*Parametric Technology Corporation* o corporación de tecnología paramétrica).

Su principal punto fuerte es el reconocimiento de marcadores, unas imágenes en 2D (dos dimensiones) con un patrón distintivo, sobre los que se apoyan para crear experiencias de realidad virtual, incluso en condiciones de poca luz o con el marcador parcialmente cubierto. También puede reconocer sencillos objetos en 3D (tres dimensiones), como cubos, y pueden definir lo que se conoce como *smart terrain* (terreno inteligente) para distinguir el tamaño y forma de una superficie, así como detectar objetos que tenga esa superficie encima.

2.3.2 Metaio

Metaio fue fundada en 2003, pero hasta 2005 no sacó su primera aplicación de realidad aumentada orientada a los usuarios [16]. Debido a lo avanzada que estaba su tecnología, sin embargo, fue adquirida por Apple en 2015 y todos sus productos fueron cesados, lo que no permitió utilizarlo para el proyecto.

Algunas de las ventajas de Metaio son un alto nivel de abstracción para el desarrollo de aplicaciones de realidad aumentada, así como soporte para diversos tipos de archivos de modelos 3D de objetos que permite hacer más fácil la importación de dichos objetos al mundo virtual.

2.3.3 Wikitude

Esta plataforma fue creada en 2008 con el objetivo de crear aplicaciones de realidad aumentada basadas en la geolocalización, sin embargo en 2012 lanzó el Wikitude SDK, el cual incluye más funcionalidades como reconocimiento de imágenes y seguimiento.

Wikitude destaca en la localización, teniendo aspectos como indicadores de dirección y localización de un marcador, y también soporta el uso de animaciones.

2.3.4 Comparación

Para poder comparar la funcionalidad de estas tres plataformas, vamos a realizar una tabla exponiendo todas las posibles funcionalidades y ventajas de todas ellas [17].

	Prueba gratuita	Reconocimiento de imágenes	Soporte para Unity	Soporte para iOS y Android	Soporte para Epson Moverio	Reconocimiento de objetos 3D	Disponible a fecha de 2016
Vuforia	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Metaio	Sí	Sí	Sí	Sí	Sí	Sí	No
Wikitude	Sí	Sí	Sí	Sí	Sí	Sí	Sí

Tabla 1. Comparación de plataformas de realidad aumentada

Como se puede observar, a nivel de funcionalidad todos los SDK cumplen con los requisitos básicos que se tuvieron en cuenta para decidir la tecnología a usar. Metaio dejó de estar disponible en 2015, lo que hace que fuera imposible utilizarlo.

Al final el SDK elegido fue Vuforia debido a que la extensión para Unity de Wikitude se encontraba en fase beta al comienzo de este proyecto, lo cual la hacía más arriesgada en caso de que hubiera un problema durante el desarrollo de la aplicación.

3. ANÁLISIS

Para iniciar el proceso del desarrollo de *software*, lo primero que hay que tener claro son las características que tiene que tener nuestra aplicación. Para ello, primero esbozaremos los llamados requisitos del usuario, que listan todos los requerimientos que deseamos que nuestra aplicación sea capaz de realizar, desde el punto de vista de un usuario o cliente.

3.1 Elección de juegos

A fin de realizar una aplicación sencilla que permitiera explorar diversos aspectos de la realidad aumentada, la decisión tomada sobre los juegos que crear en este proyecto fue la adaptación de clásicos de los videojuegos.

El primer videojuego que se va a adaptar a la realidad aumentada es el conocido Breakout o Arkanoid. Creado en 1970 por la compañía Atari (y derivado en 1986 a su versión más popular, Arkanoid, por la empresa Taito), Breakout es un juego que consiste en desplazar una barra que hace rebotar una pelota cuando choca con ella, intentando que la pelota no llegue más lejos que la barra. Cada nivel es completado cuando la pelota destruye una serie de bloques situados en la parte superior de la pantalla rebotando con ellos. Un ejemplo de situación en este juego se muestra en la figura 17.



Figura 17. Escena de juego de Breakout

El motivo por el que se decidió adaptar este juego a la realidad aumentada es por la simplicidad de su mecánica de juego, que permite centrarse en buscar una forma de control más innovadora para este conocido clásico.

El segundo juego, a fin de no realizar solamente adaptaciones, es un juego original basado en la famosa aplicación de laberinto para *smartphones*, mostrado en la figura 18.



Figura 18. Escena del juego *Labyrinth* para iOS

Esta aplicación consiste en que, a través de la inclinación del dispositivo, se permita mover una bola a través de un laberinto creado en la pantalla del *smartphone* para evitar agujeros y llegar a una meta. La inclinación simula un efecto de gravedad en la pelota, permitiendo así el desplazamiento de la misma.

De esta aplicación se ha cogido el concepto del manejo de una pelota a través de la gravedad, y por ello se decidió crear un juego que consiste en, inclinando el tablero de juego en el mundo real, evitar que una pelota sea aplastada por unas cajas que caen de encima del tablero.

La diferencia con el juego del laberinto, aparte del objetivo del mismo, es que el dispositivo en el que se muestra el juego no es el mismo que el que debe inclinarse para manejarlo. En el caso de este proyecto, no es necesario el uso de otro dispositivo para el manejo del juego sino que es necesario una imagen distintiva, o marcador, para hacer las veces de tablero de juego.

3.2 Requisitos del usuario

Los requisitos del usuario, según una serie de criterios como la concreción de estos o al elemento de la aplicación a la que se refieren, serán divididos en diferentes secciones:

3.2.1 Capacidades generales

Las características generales que debe cumplir la aplicación son las siguientes:

- La aplicación proporcionará al usuario la oportunidad de jugar a distintos juegos con controles intuitivos del mundo real. Cada uno de los juegos de los que disponga la aplicación explorarán una forma de control distinta, que haga que la experiencia de juego sea única.
- La aplicación debe ser fácil e intuitiva, con controles sencillos y que permitan al usuario dejar de darse cuenta de que los objetos virtuales no se encuentran ahí en realidad, para así aumentar su experiencia.

3.2.2 Restricciones generales

Las restricciones a las que se enfrenta la aplicación son:

- La aplicación se limita únicamente a añadir elementos virtuales a la visión del usuario. La aplicación, en este caso, no permite alterar en ninguna forma los elementos del mundo real.
- Los elementos virtuales mostrados a través de la aplicación siempre aparecen delante de los elementos del mundo real, lo que puede impedir la visión de algunos elementos reales.
- Debido a la limitación de la tecnología, algunas figuras pueden no ser reconocidas inmediatamente por la aplicación debido a factores como su forma o silueta o a otras condiciones externas como la iluminación.

3.2.3 Características del usuario

Los requisitos que debe cumplir un usuario para poder utilizar los juegos de la aplicación son los siguientes:

- El usuario debe tener una edad superior a 13 años para el manejo de la aplicación, debido a que supone interacciones con el mundo real con las que hay que ser responsable.
- El usuario debe tener un conocimiento básico del funcionamiento del dispositivo en el que se ejecuta la aplicación y el sistema operativo de este, para poder abrir e instalar la aplicación y poder seguir las instrucciones de uso de esta.

- El usuario debe tener un conocimiento mínimo del idioma español, dado que algunas instrucciones se encontrarán en este idioma.
- El usuario no necesita ninguna clase de conocimiento técnico del desarrollo de la aplicación para poder disfrutar de su uso.

3.2.4 Entorno operativo

Las características necesarias en el dispositivo para que la aplicación funcione correctamente son:

- El dispositivo debe tener memoria suficiente para poder instalar la aplicación y ejecutarla sin que esta reduzca o vea afectado de alguna forma su rendimiento.
- El dispositivo debe disponer de una cámara para obtener información gráfica del mundo real, y así poder utilizar dicha información en la aplicación para interactuar con ella.
- El dispositivo debe disponer de una pantalla para poder mostrar en ella los elementos virtuales de la aplicación.
- El dispositivo debe disponer de uno de los siguientes sistemas operativos instalado: Android o iOS. Esto es debido al gran alcance de usuarios que tienen ambos.

3.2.5 Entorno de seguridad

Al no disponer de ninguna clase de cuenta en la que se requieran credenciales de ningún tipo, el robo de información del usuario no es una preocupación para el desarrollo de la aplicación.

El acceso a la aplicación únicamente estará restringido por las medidas de seguridad externas que pueda aplicar el usuario, ya sean propias del sistema operativo o de aplicaciones ajenas a esta.

3.2.6 Suposiciones y dependencias

Las siguientes suposiciones deben estar cumplidas por el usuario para posibilitar el uso de la aplicación:

- El usuario debe haber aceptado los términos de servicio y condiciones de la tienda virtual correspondiente a su dispositivo (Play Store en el caso de Android, App Store en el caso de iOS) a través de la cual se debe descargar la aplicación para su instalación.
- El usuario debe conceder los permisos necesarios a la aplicación para poder funcionar correctamente, en caso contrario algunas funcionalidades, como la cámara, podrían verse inutilizadas.

3.2.7 Requisitos específicos

A continuación se van a enumerar, en formato de tablas, una lista más concreta de requisitos específicos que debe cumplir la aplicación.

Estos requisitos aparecerán divididos en dos categorías, donde los requisitos de capacidades representan lo que la aplicación debe ser capaz de realizar, y los requisitos de restricciones representan limitaciones que la aplicación debe marcar para su funcionamiento.

El formato que seguirán estas tablas es el siguiente:

- **ID:** identificador del requisito, con el formato UCXX para los requisitos de capacidades y URXX para los requisitos de restricciones (donde XX representa un número de dos cifras que determina la posición del requisito en la lista).
- **Nombre:** un identificador corto y conciso de cada requisito que describe en pocas palabras su función.
- **Necesario:** campo binario en el que se marcará sí o no dependiendo de si es un requisito esencial o se puede prescindir de él.
- **Prioridad:** en el caso de no ser un requisito necesario, este campo dispondrá de tres valores posibles (alta, media o baja) para representar lo importante que es el requisito.
- **Verificabilidad:** si puede comprobarse que el requisito especificado está cumplido en la aplicación de forma absoluta o no.
- **Descripción:** resumen de las características que debe tener este requisito.

3.2.7.1 Requisitos de capacidades

ID	UC01	Nombre	Variedad
Necesario	Sí	Prioridad	-
Verificabilidad	Sí		
Descripción	Varios juegos deben estar disponibles para que el usuario pueda elegir al que prefiere jugar, desde un menú principal en el que seleccionarlo		

Tabla 2. Requisito de capacidades UC01

ID	UC02	Nombre	Dificultad
Necesario	No	Prioridad	Alta
Verificabilidad	Sí		
Descripción	El usuario debe ser capaz de seleccionar diferentes dificultades para cada uno de los juegos		

Tabla 3. Requisito de capacidades UC02

ID	UC03	Nombre	Pausa
Necesario	Sí	Prioridad	-
Verificabilidad	Sí		
Descripción	El usuario debe ser capaz de poner el juego en pausa en cualquier momento		

Tabla 4. Requisito de capacidades UC03

ID	UC04	Nombre	Cámara
Necesario	Sí	Prioridad	-
Verificabilidad	Sí		
Descripción	El juego debe poder recibir información del mundo real a través de la cámara del dispositivo		

Tabla 5. Requisito de capacidades UC04

ID	UC05	Nombre	Control
Necesario	Sí	Prioridad	-
Verificabilidad	Sí		
Descripción	Los controles del juego tienen que basarse en movimientos del mundo real		

Tabla 6. Requisito de capacidades UC05

ID	UC06	Nombre	Instrucciones
Necesario	No	Prioridad	Baja
Verificabilidad	Sí		
Descripción	El juego debe contener instrucciones sobre su objetivo y los controles para que el usuario pueda conocer las reglas de antemano		

Tabla 7. Requisito de capacidades UC06

ID	UC07	Nombre	Menú
Necesario	Sí	Prioridad	-
Verificabilidad	Sí		
Descripción	El usuario debe ser capaz de poder volver al menú principal de la aplicación en cualquier momento del juego		

Tabla 8. Requisito de capacidades UC07

ID	UC08	Nombre	Cerrar aplicación
Necesario	Sí	Prioridad	-
Verificabilidad	Sí		
Descripción	La aplicación debe proveer de medios para salir de ella en caso de querer dejar de jugar		

Tabla 9. Requisito de capacidades UC08

ID	UC09	Nombre	Progreso
Necesario	No	Prioridad	Baja
Verificabilidad	Sí		
Descripción	La aplicación debe proporcionar alguna forma de progresar en el juego, para aportar la sensación de logro al usuario		

Tabla 10. Requisito de capacidades UC09

ID	UC10	Nombre	Fin del juego
Necesario	Sí	Prioridad	-
Verificabilidad	Sí		
Descripción	En caso de perder la partida en alguno de los dos juegos, la aplicación debe proporcionar medios para volver a jugar o volver al menú de selección		

Tabla 11. Requisito de capacidades UC10

ID	UC11	Nombre	Arkanoid
Necesario	No	Prioridad	Alta
Verificabilidad	Sí		
Descripción	El juego basado en Arkanoid debe tener una mecánica lo más parecida posible al original haciendo uso de controles a través de realidad aumentada		

Tabla 12. Requisito de capacidades UC11

ID	UC12	Nombre	Resultados
Necesario	Sí	Prioridad	-
Verificabilidad	Sí		
Descripción	Ambos juegos deben ofrecer la posibilidad de ganar o perder la partida, y la posibilidad de que cualquiera de estos dos sucesos ocurra debe variar dependiendo de la dificultad seleccionada por el usuario		

Tabla 13. Requisito de capacidades UC12

ID	UC13	Nombre	Relación de controles
Necesario	No	Prioridad	Media
Verificabilidad	Sí		
Descripción	Los controles para cada uno de los juegos deben estar relacionados de alguna forma con la mecánica del juego para que resulte natural		

Tabla 14. Requisito de capacidades UC13

3.2.7.2 Requisitos de restricciones

ID	UR01	Nombre	Visibilidad
Necesario	No	Prioridad	Media
Verificabilidad	Sí		
Descripción	Los elementos virtuales deben ocupar lo menos posible en el campo de visión del mundo real, para no reducir la visibilidad del usuario		

Tabla 15. Requisito de restricciones UR01

ID	UR02	Nombre	Idioma
Necesario	Sí	Prioridad	-
Verificabilidad	Sí		
Descripción	La aplicación solo estará disponible en el idioma español		

Tabla 16. Requisito de restricciones UR02

ID	UR03	Nombre	Interacción
Necesario	No	Prioridad	Alta
Verificabilidad	Sí		
Descripción	La aplicación no debe requerir interacción física del usuario con el dispositivo		

Tabla 17. Requisito de restricciones UR03

ID	UR04	Nombre	Velocidad de respuesta
Necesario	No	Prioridad	Media
Verificabilidad	No		
Descripción	La aplicación debe iniciarse en menos de 5 segundos, y responder al usuario inmediatamente para permitir la experiencia inmersiva		

Tabla 18. Requisito de restricciones UR04

ID	UR05	Nombre	Identificación de dispositivo
Necesario	Sí	Prioridad	-
Verificabilidad	Sí		
Descripción	La aplicación debe poder detectar el dispositivo en el que se está ejecutando para variar acordemente		

Tabla 19. Requisito de restricciones UR05

ID	UR06	Nombre	Intuitividad
Necesario	No	Prioridad	Baja
Verificabilidad	No		
Descripción	El uso de la aplicación debe ser intuitivo, con controles sencillos y representaciones gráficas fáciles de entender		

Tabla 20. Requisito de restricciones UR06

3.3 Casos de uso

Una vez obtenidos los requisitos que queremos que cumpla nuestra aplicación, el siguiente paso fue diseñar unos sencillos casos de uso. Estos pretenden mostrar, de manera sencilla en forma de diagrama, ejemplos de situaciones que se pueden dar al hacer uso de la aplicación.

Para explicar los diferentes escenarios que pueden surgir, se han creado tres casos de uso diferentes. Estos, representados en las figuras 19, 20 y 21, muestran la elección de uno de los juegos al que empezar a jugar, el desarrollo de una partida del juego Arkanoid y el desarrollo de una partida del juego del laberinto, respectivamente.

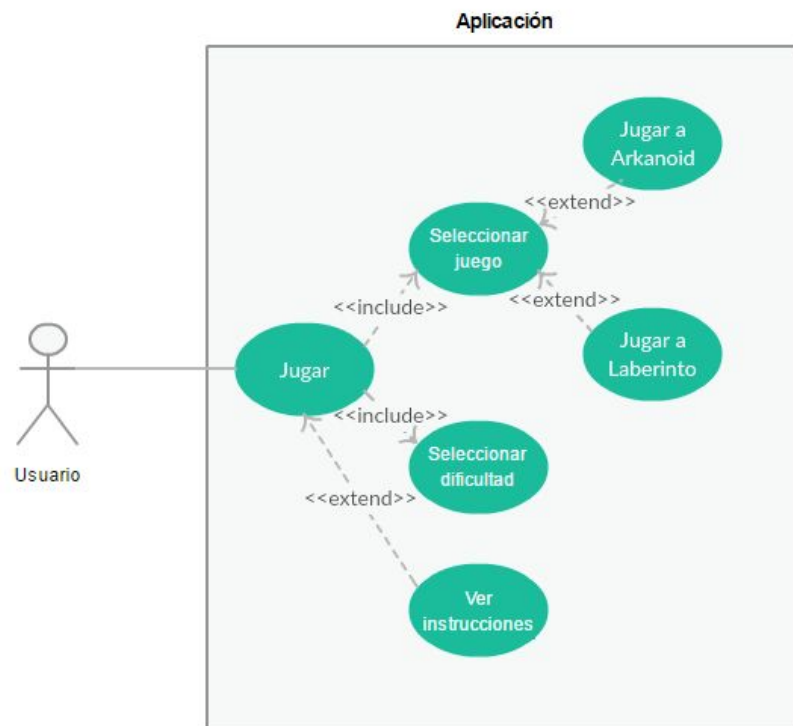


Figura 19. Caso de uso de selección de juego

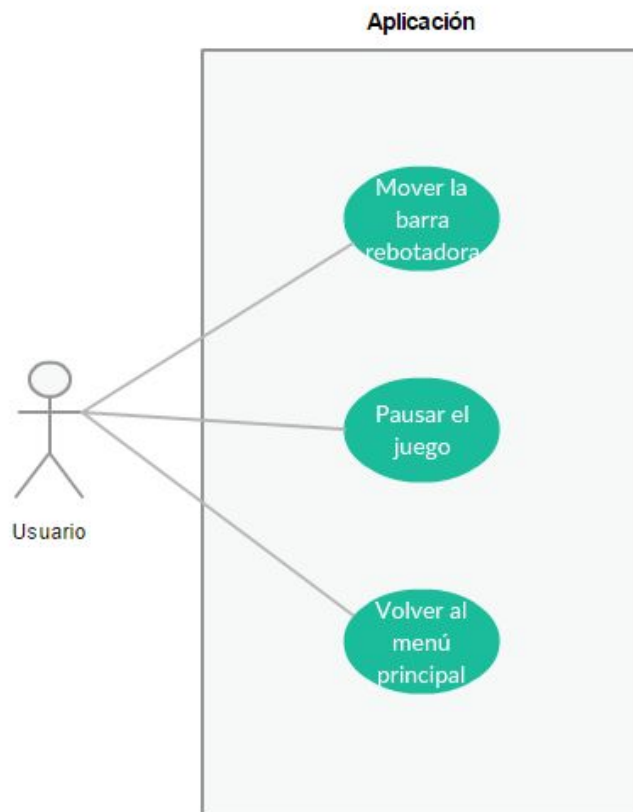


Figura 20. Caso de uso de manejo de Arkanoid

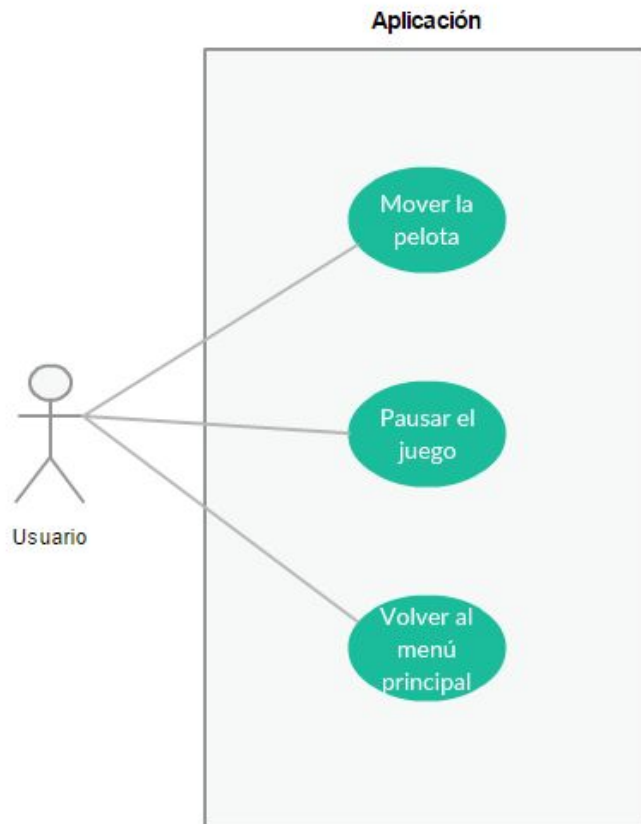


Figura 21. Caso de uso de manejo del juego del laberinto

Para poder entender mejor estos diagramas, a continuación se expondrán en unas tablas estos escenarios con posibles variantes de las situaciones y un desarrollo lineal de las acciones. Estas tablas constan de los siguientes apartados:

- **Nombre:** breve nombre descriptivo del escenario.
- **Descripción:** resumen del objetivo de la acción, así como de otras características que definen el escenario.
- **Actores:** todas las personas involucradas en la acción.
- **Precondiciones:** la situación y condiciones que deben cumplirse para que el escenario pueda suceder y la acción pueda ocurrir.
- **Flujo básico:** desarrollo principal de la acción, el que tiene más probabilidades de ocurrir de forma natural con todos los pasos descritos y enumerados.
- **Flujo alternativo:** si uno de los pasos descritos en el punto anterior puede llevar a un desarrollo diferente de la acción estará especificado en esta sección, así como el desarrollo de ese flujo alternativo del escenario.

- **Flujo de excepción:** si uno de los pasos descritos en el flujo básico puede llevar al usuario a una situación en la que no puede cumplir el objetivo que se había planteado inicialmente, estará explicado en este punto así como la situación que tiene que darse para interrumpir el flujo principal.
- **Postcondiciones:** la situación y condiciones que se pueden encontrar después de haber completado los flujos principal y alternativos, indicativos de que la acción se ha completado satisfactoriamente.

Nombre	Seleccionar juego
Descripción	Este escenario describe cómo se puede elegir uno de los dos juegos para probar, desde el inicio de la aplicación hasta que se han seleccionado factores como el juego o la dificultad del mismo
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> ● La aplicación está iniciada ● El usuario tiene el dispositivo colocado ● El usuario se encuentra en el menú de selección de juego (menú principal)
Flujo básico	<ol style="list-style-type: none"> 1. El usuario selecciona el juego Arkanoid 2. El usuario elige la dificultad fácil 3. El juego comienza
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario selecciona el juego del laberinto 2. El usuario selecciona la dificultad fácil 3. El juego comienza
Flujo alternativo	<ol style="list-style-type: none"> 2. El usuario elige la dificultad media 3. El juego comienza
Flujo alternativo	<ol style="list-style-type: none"> 2. El usuario elige la dificultad difícil 3. El juego comienza
Flujo alternativo	<ol style="list-style-type: none"> 2. El usuario lee las instrucciones del juego 3. El usuario elige la dificultad fácil 4. El juego comienza
Flujo de excepción	1, 2. La aplicación es cerrada
Postcondiciones	<ul style="list-style-type: none"> ● La aplicación está iniciada ● El usuario tiene el dispositivo colocado ● El juego y la dificultad seleccionadas se cargan

Tabla 21. Caso de uso de selección de juego

Nombre	Manejo de Arkanoid
Descripción	Este escenario describe el manejo del juego Arkanoid, así como posibles situaciones que se pueden dar durante el mismo, hasta que la partida termine en una victoria o una derrota
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • La aplicación está iniciada • El usuario tiene el dispositivo colocado • El usuario se encuentra en el juego Arkanoid
Flujo básico	<ol style="list-style-type: none"> 1. El usuario mueve la barra rebotadora 2. El usuario evita que la pelota caiga por debajo de la barra rebotadora 3. Todos los bloques son destruidos y el usuario gana la partida
Flujo alternativo	<ol style="list-style-type: none"> 2. La pelota cae por debajo de la barra rebotadora
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario pausa el juego 2. El usuario reanuda el juego 3. El usuario mueve la barra rebotadora 4. El usuario evita que la pelota caiga por debajo de la barra rebotadora 5. Todos los bloques son destruidos y el usuario gana la partida
Flujo alternativo	<ol style="list-style-type: none"> 2. El usuario pausa el juego 3. El usuario reanuda el juego 4. El usuario evita que la pelota caiga por debajo de la barra rebotadora 5. Todos los bloques son destruidos y el usuario gana la partida
Flujo de excepción	1, 2. El usuario elige volver al menú principal
Flujo de excepción	1, 2. La aplicación es cerrada
Postcondiciones	<ul style="list-style-type: none"> • La aplicación está iniciada • El usuario tiene el dispositivo colocado • El usuario ha finalizado el nivel

Tabla 22. Caso de uso de manejo de Arkanoid

Nombre	Manejo del juego del laberinto
Descripción	Este escenario describe el manejo del juego del laberinto, así como posibles situaciones que se pueden dar durante el mismo, hasta que la partida termine en una victoria o una derrota
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • La aplicación está iniciada • El usuario tiene el dispositivo colocado • El usuario se encuentra en el juego del laberinto
Flujo básico	<ol style="list-style-type: none"> 1. El usuario mueve la pelota 2. El usuario evita que la pelota caiga fuera de los límites del tablero o debajo de una caja 3. Todas las cajas caen sin aplastar la pelota y el usuario gana la partida
Flujo alternativo	2. La pelota es aplastada por una caja
Flujo alternativo	2. La pelota cae fuera de los límites del tablero
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario pausa el juego 2. El usuario reanuda el juego 3. El usuario mueve la pelota 4. El usuario evita que la pelota caiga fuera de los límites del tablero o debajo de una caja 5. Todas las cajas caen sin aplastar la pelota y el usuario gana la partida
Flujo alternativo	<ol style="list-style-type: none"> 2. El usuario pausa el juego 3. El usuario reanuda el juego 4. El usuario evita que la pelota caiga fuera de los límites del tablero o debajo de una caja 5. Todas las cajas caen sin aplastar la pelota y el usuario gana la partida
Flujo de excepción	1, 2. La aplicación es cerrada
Flujo de excepción	1, 2. El usuario elige volver al menú principal
Postcondiciones	<ul style="list-style-type: none"> • La aplicación está iniciada • El usuario tiene el dispositivo colocado • El usuario ha finalizado el nivel

Tabla 23. Caso de uso de manejo del juego del laberinto

3.4 Requisitos del software

A partir de los requisitos del usuario descritos encima se han redactado los siguientes requisitos del *software*, que describen las medidas que se han decidido aplicar al *software* para que cumpla con los requisitos del usuario de una forma más técnica.

Los requisitos estarán escritos en un formato de tablas. El formato que seguirán estas tablas es el siguiente:

- **ID:** identificador del requisito, con el formato SFXX para los requisitos funcionales, SDXX para los requisitos de desempeño, SOXX para los requisitos operacionales, SRXX para los requisitos de recursos, SSXX para los requisitos de seguridad y SNXX para los requisitos no funcionales (donde XX representa un número de dos cifras que determina la posición del requisito en la lista).
- **Nombre:** un identificador corto y conciso de cada requisito que describe en pocas palabras su función.
- **Necesario:** campo binario en el que se marcará sí o no dependiendo de si es un requisito esencial o se puede prescindir de él.
- **Prioridad:** en el caso de no ser un requisito necesario, este campo dispondrá de tres valores posibles (alta, media o baja) para representar lo importante que es el requisito.
- **Verificabilidad:** si puede comprobarse que el requisito especificado está cumplido en la aplicación de forma absoluta o no.
- **Requisito del usuario:** el requisito del usuario al que este requisito del *software* pone solución y con el que está asociado.
- **Descripción:** resumen de las características que debe tener este requisito.

Los requisitos del *software*, según una serie de criterios como la concreción de estos o al elemento de la aplicación a la que se refieren, serán divididos en diferentes secciones:

3.4.1 Requisitos funcionales

ID	SF01	Nombre	Pausa
Necesario	Sí	Prioridad	-
Verificabilidad	Sí	Requisito del usuario	UC03
Descripción	El sistema permitirá poner en pausa el juego en cualquier momento		

Tabla 24. Requisito funcional SF01

ID	SF02	Nombre	Instrucciones
Necesario	No	Prioridad	Baja
Verificabilidad	Sí	Requisito del usuario	UC06
Descripción	El sistema permitirá mostrar unas instrucciones para que el usuario pueda conocer las reglas de antemano		

Tabla 25. Requisito funcional SF02

ID	SF03	Nombre	Menú
Necesario	Sí	Prioridad	-
Verificabilidad	Sí	Requisito del usuario	UC07
Descripción	El sistema permitirá al usuario volver al menú principal en cualquier momento del juego		

Tabla 26. Requisito funcional SF03

ID	SF04	Nombre	Cerrar aplicación
Necesario	Sí	Prioridad	-
Verificabilidad	Sí	Requisito del usuario	UC08
Descripción	El sistema permitirá al usuario salir de la aplicación en caso de querer dejar de jugar		

Tabla 27. Requisito funcional SF04

ID	SF05	Nombre	Fin del juego
Necesario	Sí	Prioridad	-
Verificabilidad	Sí	Requisito del usuario	UC10
Descripción	El sistema proporcionará medios para volver al menú principal en caso de que el usuario pierda la partida		

Tabla 28. Requisito funcional SF05

3.4.2 Requisitos de desempeño

ID	SD01	Nombre	Velocidad de respuesta
Necesario	No	Prioridad	Media
Verificabilidad	No	Requisito del usuario	UR04
Descripción	El sistema se asegurará de que la aplicación se inicia en menos de 5 segundos, y que esta responderá inmediatamente al usuario para permitir la experiencia inmersiva		

Tabla 29. Requisito de desempeño SD01

3.4.3 Requisitos operacionales

ID	SO01	Nombre	Variedad
Necesario	Sí	Prioridad	-
Verificabilidad	Sí	Requisito del usuario	UC01
Descripción	El menú principal proveerá la posibilidad de elegir los distintos juegos a los que jugar		

Tabla 30. Requisito operacional SO01

ID	SO02	Nombre	Dificultad
Necesario	No	Prioridad	Alta
Verificabilidad	Sí	Requisito del usuario	UC02
Descripción	El menú principal proveerá la posibilidad de elegir distintas dificultades para cada juego		

Tabla 31. Requisito operacional SO02

ID	SO03	Nombre	Idioma
Necesario	Sí	Prioridad	-
Verificabilidad	Sí	Requisito del usuario	UR02
Descripción	La aplicación estará disponible en español		

Tabla 32. Requisito operacional SO03

3.4.4 Requisitos de recursos

ID	SR01	Nombre	Cámara
Necesario	Sí	Prioridad	-
Verificabilidad	Sí	Requisito del usuario	UC04
Descripción	El sistema recibirá información para permitir el juego a través de la cámara del dispositivo		

Tabla 33. Requisito de recursos SR01

ID	SR02	Nombre	Interacción
Necesario	No	Prioridad	Alta
Verificabilidad	Sí	Requisito del usuario	UR03
Descripción	La aplicación no requerirá datos recibidos directamente del usuario		

Tabla 34. Requisito de recursos SR02

ID	SR03	Nombre	Identificación de dispositivo
Necesario	Sí	Prioridad	-
Verificabilidad	Sí	Requisito del usuario	UR05
Descripción	La aplicación detectará, y variará acordeamente con, el dispositivo en el que se está ejecutando		

Tabla 35. Requisito de recursos SR03

3.4.5 Requisitos de seguridad

ID	SS01	Nombre	Visibilidad
Necesario	No	Prioridad	Media
Verificabilidad	Sí	Requisito del usuario	UR01
Descripción	El sistema asegurará que los elementos virtuales ocuparán lo menos posible en el campo de visión del mundo real, para no reducir la visibilidad del usuario		

Tabla 36. Requisito de seguridad SS01

3.4.6 Requisitos no funcionales

ID	SN01	Nombre	Control
Necesario	Sí	Prioridad	-
Verificabilidad	Sí	Requisito del usuario	UC05
Descripción	Los controles del juego estarán basados en movimientos del mundo real		

Tabla 37. Requisito no funcional SN01

ID	SN02	Nombre	Progreso
Necesario	No	Prioridad	Baja
Verificabilidad	Sí	Requisito del usuario	UC09
Descripción	El sistema proporcionará al usuario la sensación de progreso a través del cambio de niveles		

Tabla 38. Requisito no funcional SN02

ID	SN03	Nombre	Arkanoid
Necesario	No	Prioridad	Alta
Verificabilidad	Sí	Requisito del usuario	UC11
Descripción	La mecánica de juego de Arkanoid será igual que la del juego original, variando el control a través de la realidad aumentada		

Tabla 39. Requisito no funcional SN03

ID	SN04	Nombre	Resultados
Necesario	Sí	Prioridad	-
Verificabilidad	Sí	Requisito del usuario	UC12
Descripción	El sistema asegurará que se pueda ganar y perder en ambos juegos, variando la dificultad de cada caso con la dificultad del juego seleccionada		

Tabla 40. Requisito no funcional SN04

ID	SN05	Nombre	Relación de controles
Necesario	No	Prioridad	Media
Verificabilidad	Sí	Requisito del usuario	UC13
Descripción	El sistema asegurará que los controles de ambos juegos le resultarán naturales para el usuario		

Tabla 41. Requisito no funcional SN05

ID	SN06	Nombre	Intuitividad
Necesario	No	Prioridad	Baja
Verificabilidad	No	Requisito del usuario	UR06
Descripción	Las representaciones gráficas y los controles serán sencillos de entender para el usuario		

Tabla 42. Requisito no funcional SN06

3.4.7 Matriz de trazabilidad

	U C 0 1	U C 0 2	U C 0 3	U C 0 4	U C 0 5	U C 0 6	U C 0 7	U C 0 8	U C 0 9	U C 1 0	U C 1 1	U C 1 2	U C 1 3	U R 0 1	U R 0 2	U R 0 3	U R 0 4	U R 0 5	U R 0 6
SF01			X																
SF02						X													
SF03							X												
SF04								X											
SF05										X									
SD01																	X		
SO01	X																		
SO02		X																	
SO03															X				
SR01				X															
SR02																X			
SR03																		X	
SS01														X					
SN01					X														
SN02									X										
SN03											X								
SN04												X							
SN05													X						
SN06																			X

Tabla 43. Matriz de trazabilidad

4. DISEÑO

Una vez especificados los requisitos que debe cumplir la aplicación se procedió al diseño de la misma. Para ello, se van a exponer a continuación distintas fases del proceso a través de diagramas, tablas e imágenes.

El proceso comienza con el diseño de la arquitectura del sistema. Una vez completado, se procede con conceptos de la implementación e interfaz de la aplicación.

4.1 Arquitectura

Para el diseño de la arquitectura, se ha interpretado el proyecto como tres aplicaciones independientes interconectadas entre ellas: el menú de selección de opciones como el juego o la dificultad, el juego Arkanoid y el juego del laberinto.

Para estas últimas dos partes del proyecto se realizó un diagrama de clases diferente (figuras 22 y 23), que describe la organización de los componentes de esa aplicación en solitario. El menú principal no se diseñó con un diagrama, debido a que sigue una arquitectura similar a la juego Arkanoid eliminando los elementos de juego y redirigiendo la aplicación en diferentes direcciones según el botón pulsado.

En estos diagramas se ha interpretado solamente la parte a implementar en la aplicación. A pesar de que las tres partes del proyecto hacen uso de recursos de Unity, como los cálculos físicos o el uso de sus componentes, estos no se han tenido en cuenta al realizar los diagramas debido a que los componentes a implementar no necesitan ser programados para hacer uso de los recursos de Unity, sino que lo hacen directamente.

Por otro lado, a pesar de que se ha utilizado la librería de Vuforia para el uso de funcionalidades de realidad aumentada no es posible acceder a los componentes internos de la misma, de forma que se ha utilizado un patrón de diseño *facade* (fachada) en el que se representa toda la librería como un único componente con el que se debe interactuar para poder hacer cálculos de realidad aumentada.

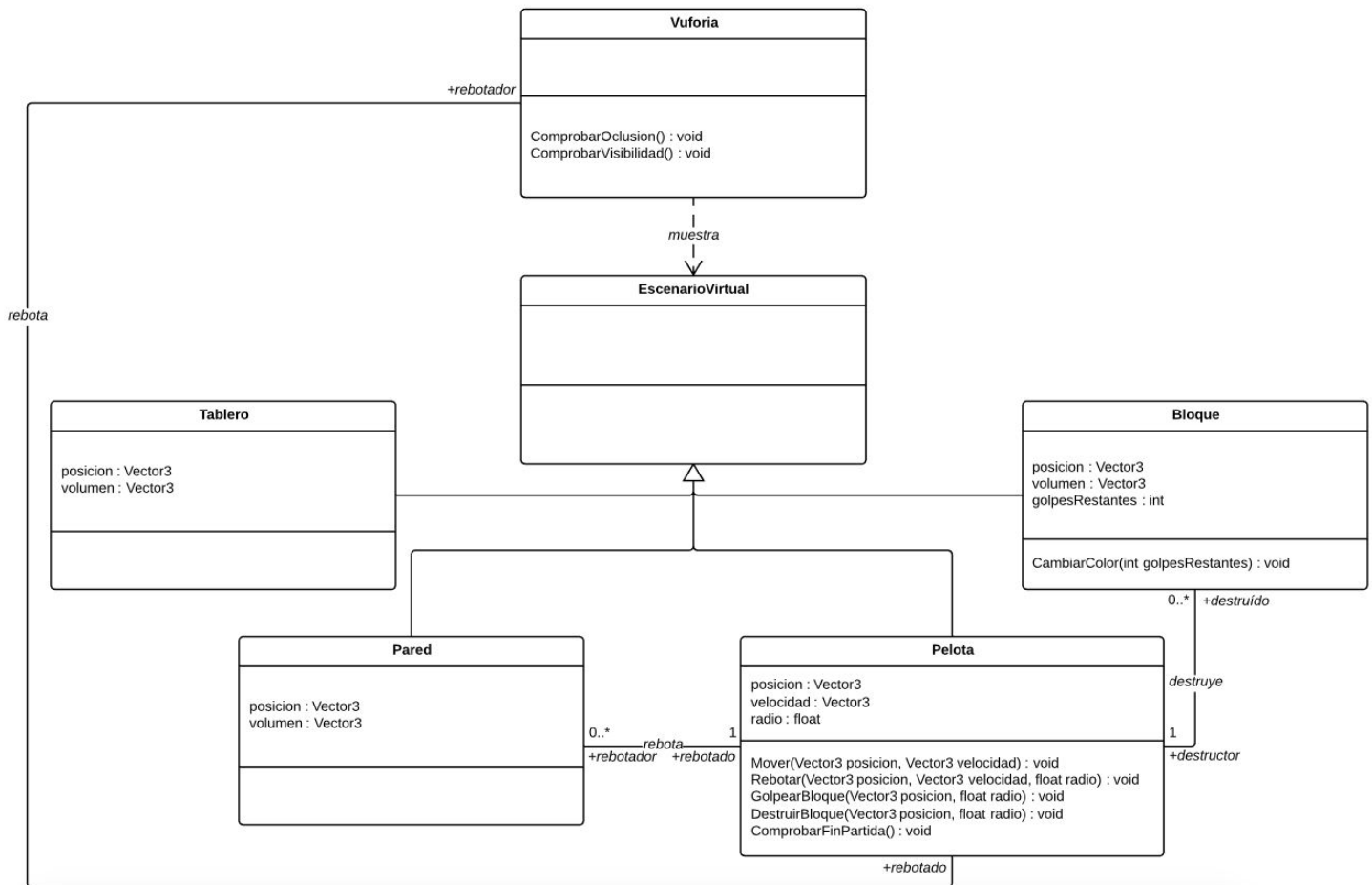


Figura 22. Diagrama de clases del juego Arkanoid

Como se puede observar, los objetos utilizados para poder jugar en este caso son todos miembros de la categoría escenario virtual. Esto es así debido a que, en caso de no detectar el marcador que debe desencadenar el juego, estos objetos desaparecen de la pantalla para evitar que el juego continúe.

Los elementos que forman parte del escenario virtual están relacionados directamente con la librería de Vuforia, encargada de la tarea del reconocimiento del marcador. Vuforia también se encarga de comprobar la oclusión del marcador con elementos del mundo real, de forma que cuando parte del escenario esté oculta se pueda comprobar si coincide con la sección en la que se encuentra la pelota (indicando así que el usuario la está tocando).

El diagrama del menú principal se correspondería con los elementos Vuforia y EscenarioVirtual de este diagrama, ya que solo se necesita el tablero y el reconocimiento de oclusión de Vuforia para poder seleccionar las opciones del juego.

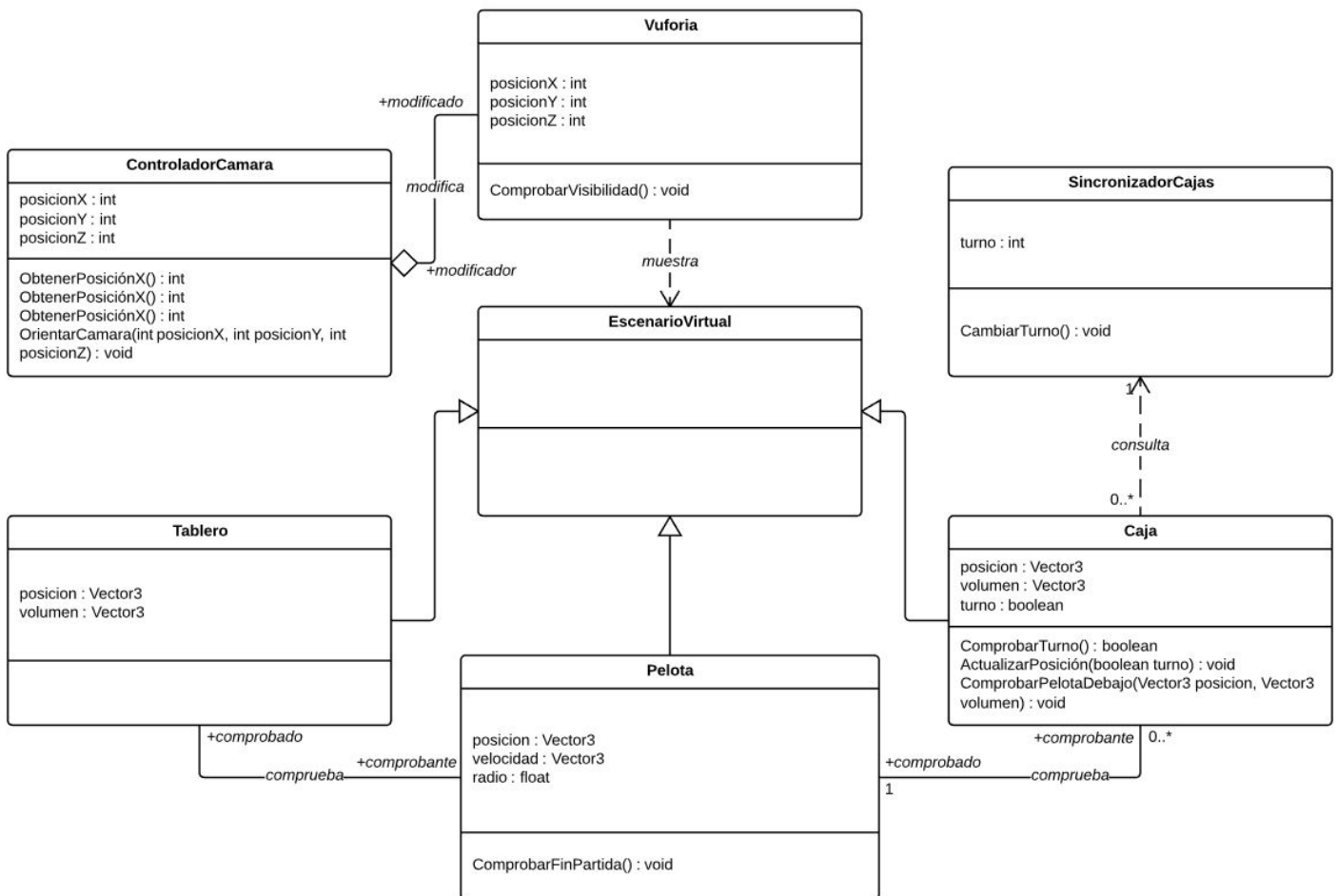


Figura 23. Diagrama de clases del juego del laberinto

En el caso del juego del laberinto, la mecánica de los elementos del escenario virtual funciona igual que en el juego Arkanoid. Se incluyen además diversos elementos, como un objeto cuya única función es sincronizar las cajas para que caigan de una en una.

La parte más importante en este diagrama es el controlador de la cámara, que se encarga de detectar los valores transmitidos por el acelerómetro del dispositivo y convertirlos al ángulo en el que se encuentra la cámara, para así poder aplicar ese valor a la cámara de Vuforia y modificar cómo afecta la gravedad al juego.

4.2 Descripción de los componentes

A continuación se procederá a describir los componentes que conforman la aplicación, así como los requisitos que ayudan a satisfacer. El formato de cada uno de los componentes será el siguiente:

- **ID:** identificador único de cada componente, con una nomenclatura en formato CMXX para los componentes del menú principal, CAXX para los del juego Arkanoid y CLXX para los del juego del laberinto (donde XX representa un número de dos cifras que indica la posición del componente en la lista).
- **Nombre:** pequeño resumen identificativo del componente.
- **Tipo:** explica la clase de componente del que se trata.
- **Propósito:** identificadores de los requisitos del software para los que este componente es creado y que pretende resolver.
- **Función:** el cometido de este componente en el conjunto de la aplicación, explicando su tarea principal y todas las acciones para las que ha sido incluido.
- **Subordinados:** incluye los identificadores de los componentes que forman parte del actual componente.
- **Dependencias:** qué clase de datos, componentes... necesita este componente para poder funcionar correctamente.

ID	CA01	Nombre	EscenarioVirtual
Tipo	Subsistema		
Propósito	<ul style="list-style-type: none"> • SS01 		
Función	Agrupar todos los componentes que deban desaparecer cuando el marcador en el que proyectar el juego no se encuentra a la vista del usuario		
Subordinados	<ul style="list-style-type: none"> • CA02 • CA03 • CA04 • CA05 • CA06 		
Dependencias	-		

Tabla 44. Componente CA01

ID	CA02	Nombre	Tablero
Tipo	Componente		
Propósito	<ul style="list-style-type: none"> • SS01 • SN03 • SN06 		
Función	Proporcionar un soporte físico a la pelota para que pueda mantenerse sobre él, permitiendo el manejo de la misma		
Subordinados	-		
Dependencias	<ul style="list-style-type: none"> • CA01 		

Tabla 45. Componente CA02

ID	CA03	Nombre	Bloque
Tipo	Componente		
Propósito	<ul style="list-style-type: none"> • SS01 • SN03 • SN04 • SN06 		
Función	Permitir que la pelota rebote en él, a la vez que impedir que la partida termine en victoria del usuario mientras el bloque siga existiendo		
Subordinados	-		
Dependencias	<ul style="list-style-type: none"> • CA01 • CA05 		

Tabla 46. Componente CA03

ID	CA04	Nombre	Pared
Tipo	Componente		
Propósito	<ul style="list-style-type: none"> • SS01 • SN03 • SN06 		
Función	Permitir que la pelota rebote en él e impedir que la pelota salga de los límites definidos para el juego		
Subordinados	<ul style="list-style-type: none"> • CA05 		
Dependencias	<ul style="list-style-type: none"> • CA01 		

Tabla 47. Componente CA04

ID	CA05	Nombre	Pelota
Tipo	Componente		
Propósito	<ul style="list-style-type: none"> • SF03 • SF05 • SS01 • SN02 • SN03 • SN04 • SN06 		
Función	Moverse por el tablero rebotando en los componentes que se encuentre, y destruir bloques para permitir al usuario ganar la partida		
Subordinados	<ul style="list-style-type: none"> • CA03 		
Dependencias	<ul style="list-style-type: none"> • CA01 • CA04 • CA06 		

Tabla 48. Componente CA05

ID	CA06	Nombre	Vuforia
Tipo	Subsistema		
Propósito	<ul style="list-style-type: none"> • SF01 • SR01 • SR02 • SN01 • SN05 • SN06 		
Función	Permitir que el tablero aparezca visible ante el usuario y comprobar cualquier parte de la imagen del marcador que haya podido ser bloqueada por un objeto, para así indicar dónde se encuentra actualmente la mano del jugador		
Subordinados	<ul style="list-style-type: none"> • CA05 		
Dependencias	<ul style="list-style-type: none"> • CA01 		

Tabla 49. Componente CA06

ID	CL01	Nombre	EscenarioVirtual
Tipo	Subsistema		
Propósito	<ul style="list-style-type: none"> • SS01 		
Función	Agrupar todos los componentes que deban desaparecer cuando el marcador en el que proyectar el juego no se encuentra a la vista del usuario		
Subordinados	<ul style="list-style-type: none"> • CL02 • CL03 • CL04 • CL07 		
Dependencias	-		

Tabla 50. Componente CL01

ID	CL02	Nombre	Tablero
Tipo	Componente		
Propósito	<ul style="list-style-type: none"> • SS01 • SN06 		
Función	Proporcionar un soporte físico a la pelota para que pueda mantenerse sobre él, permitiendo el manejo de la misma		
Subordinados	-		
Dependencias	<ul style="list-style-type: none"> • CL01 • CL03 		

Tabla 51. Componente CL02

ID	CL03	Nombre	Pelota
Tipo	Componente		
Propósito	<ul style="list-style-type: none"> • SF03 • SF05 • SS01 • SN02 • SN04 • SN06 		
Función	Ser movida por la ley de la gravedad, e indicar que la partida ha terminado si es atrapada debajo de una caja o si su posición se encuentra fuera de los límites del tablero		
Subordinados	<ul style="list-style-type: none"> • CL02 		
Dependencias	<ul style="list-style-type: none"> • CL01 • CL04 		

Tabla 52. Componente CL03

ID	CL04	Nombre	Caja
Tipo	Componente		
Propósito	<ul style="list-style-type: none"> • SF03 • SF05 • SS01 • SN04 • SN06 		
Función	Aparecer encima del tablero y desplazarse a un ritmo constante hacia él para intentar atrapar la pelota debajo y terminar así la partida en derrota del usuario		
Subordinados	<ul style="list-style-type: none"> • CL03 		
Dependencias	<ul style="list-style-type: none"> • CL01 • CL05 		

Tabla 53. Componente CL04

ID	CL05	Nombre	SincronizadorCajas
Tipo	Subprograma		
Propósito	-		
Función	Comunicarse con las cajas listas para aparecer encima del tablero para informar de qué caja debe ser la siguiente en aparecer y descender		
Subordinados	<ul style="list-style-type: none"> • CL04 		
Dependencias	-		

Tabla 54. Componente CL05

ID	CL06	Nombre	ControladorCamara
Tipo	Subprograma		
Propósito	<ul style="list-style-type: none"> • SN01 • SN05 • SN06 		
Función	Recopilar información del dispositivo para comprobar su orientación y posición, y modificar la cámara virtual de Vuforia para permitir ajustar la gravedad		
Subordinados	<ul style="list-style-type: none"> • CL07 		
Dependencias	-		

Tabla 55. Componente CL06

ID	CL07	Nombre	Vuforia
Tipo	Subsistema		
Propósito	<ul style="list-style-type: none"> • SF01 • SR01 • SR02 • SN01 • SN05 • SN06 		
Función	Permitir que el tablero aparezca visible ante el usuario		
Subordinados	-		
Dependencias	<ul style="list-style-type: none"> • CL01 • CL06 		

Tabla 56. Componente CL07

4.3 Conceptos

Después de decidir cómo se iba a implementar a nivel de arquitectura la aplicación, solo quedaba por diseñar la temática y otros conceptos gráficos de la misma.

La primera decisión fue elegir la imagen que servía como marcador para que el juego pudiera desarrollarse encima de ella. Debido a que el reconocimiento del mismo es totalmente manejado por la librería de Vuforia, se siguieron sus recomendaciones para la imagen provisional [18]: una imagen rica en detalles, con buen contraste y sin patrones repetitivos.

Siguiendo los consejos anteriores, al final se optó por utilizar una de las imágenes propuestas en la propia galería de Vuforia, para optimizar lo máximo posible el proceso de reconocimiento de la imagen. Como se puede observar en la figura 24, esta imagen cuenta con todas las condiciones descritas anteriormente.



Figura 24. Marcador elegido para su uso con Vuforia

Con ese diseño elegido, se procedió a diseñar cómo debía verse el menú principal. En este caso, las opciones debían aparecer de una forma clara para el usuario ocupando el menor espacio posible del mundo real, y por ello se eligió situar las opciones encima de la imagen, como si sobresalieran perpendicularmente a esta (figura 25).

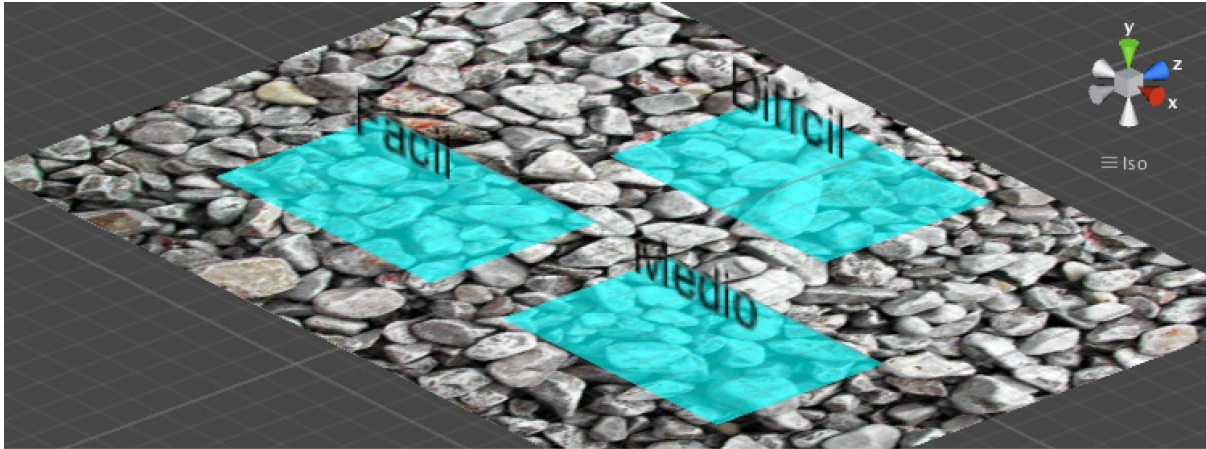


Figura 25. Posicionamiento inicial del texto con las opciones

Sin embargo, reflexionando sobre ello se llegó a la conclusión de que la mayoría de los usuarios iban a mirar al menú principal desde un ángulo de 45 grados hacia abajo, debido a que es la posición que resulta más natural. Por ello se cambió el diseño a unas palabras que sobresalieran del papel en un ángulo de 45 grados hacia arriba, como se muestra en la figura 26, permitiendo su lectura tanto desde un ángulo paralelo a la imagen como desde uno perpendicular, con todos los ángulos intermedios.

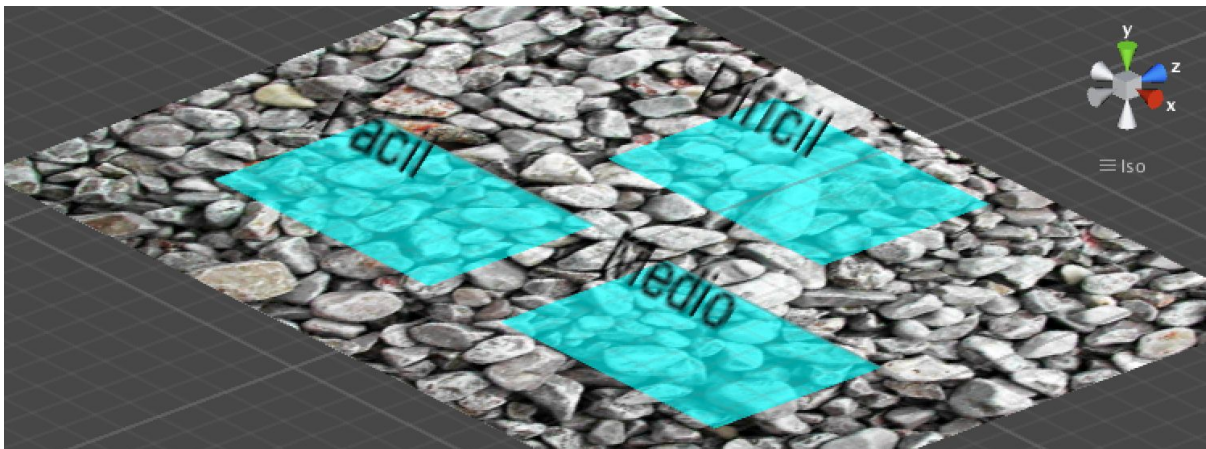


Figura 26. Posicionamiento final del texto con las opciones

En cuanto al diseño del juego Arkanoid, se intentó mantener lo más parecido posible al juego original. Los bloques, dependiendo del número de golpes que les quedan para que sean destruidos, cambian de color, y se eligieron los colores verde (1 golpe, fácil), azul (2 golpes, medio) y rojo (3 golpes, difícil) para representar lo complicado que es hacer desaparecer ese bloque por completo, basado únicamente en el número de golpes.

Debido al hecho de que no se usa el reconocimiento de la mano sino que se utiliza la oclusión de la imagen para determinar la posición de la misma, no fue posible insertar un tablero virtual en el juego. Esto es debido a que si la mano fuera reconocida, podría superponerse su área al tablero virtual creado.

Pero al no hacer esto activamente, lo virtual siempre aparece superpuesto sobre lo real. Al colocar un tablero en el escenario, a pesar de que la mano seguiría siendo reconocida para hacer rebotar la pelota, el usuario no podría verla por estar oculta tras el tablero, y el control sería más inexacto y complicado para él.

Estos motivos causaron que el tablero de juego, así como las paredes del tablero (debido a que cortaban la mano del usuario) fueran diseñados como invisibles para el usuario, a pesar de que siguen haciendo rebotar la pelota y manteniéndola en juego.

Por último, en el caso del juego del laberinto, la mano no juega un papel activo relevante para el control del juego, con lo cual se pudieron situar tanto un tablero como paredes para que el usuario pudiera ver con facilidad los límites del juego.

Al no ser un juego ya existente, se barajaron varias posibilidades para la temática del mismo. Debido a la simplicidad estética y a la multitud de texturas disponibles gratis para ello, se eligió una temática basada en los desiertos del Oeste americano. Decidido esto, las cajas se cubrieron con una textura para hacerlas parecer de madera, el tablero se modificó para parecer un desierto de arena, las paredes se diseñaron como verjas de madera y la pelota pasó a ser una hierba rodadora, como se muestra en la figura 27. Por este motivo, el juego pasó a llamarse Hierba Rodadora.

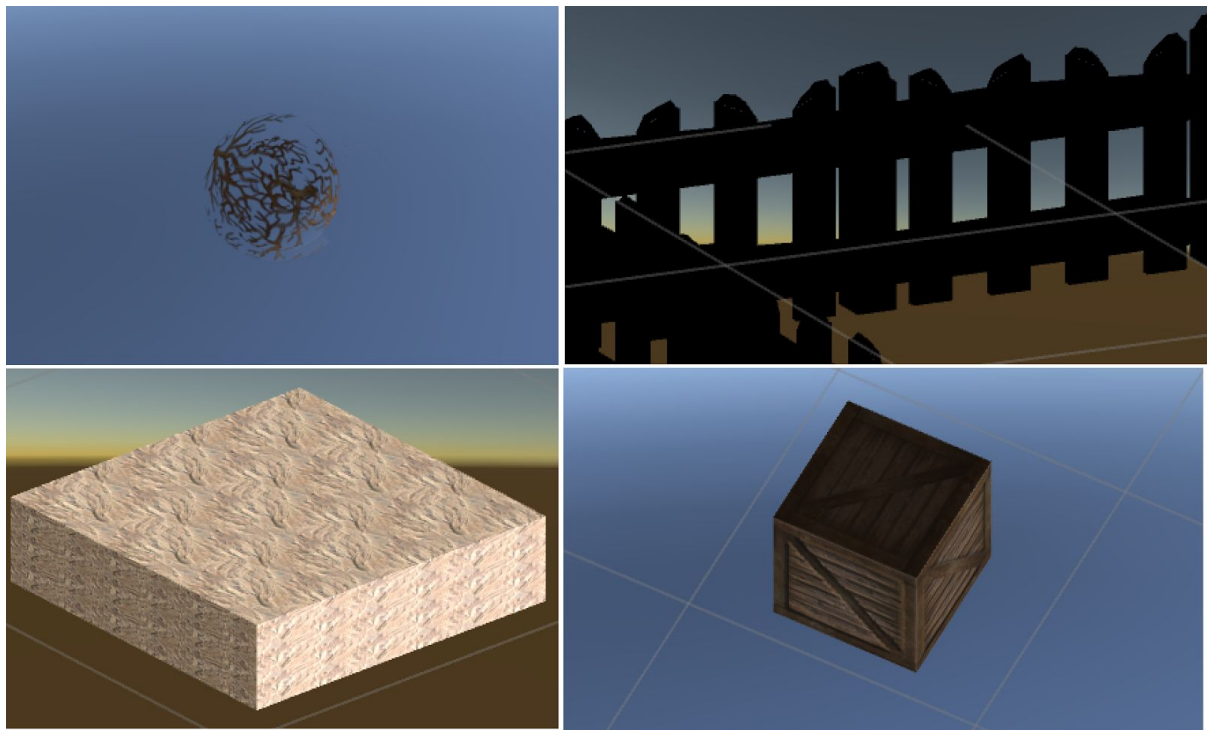


Figura 27. Diseño de la pelota (arriba izquierda), paredes (arriba derecha), suelo (abajo izquierda) y cajas (abajo derecha) del juego Hierba Rodadora

5. IMPLEMENTACIÓN

5.1 Desarrollo de realidad aumentada

Antes de decidir cómo se iba a utilizar la realidad aumentada para este proyecto, se realizaron diversas aplicaciones para probar distintas funcionalidades de Vuforia que luego pudieran integrarse con el control de los juegos.

La primera consistió en conseguir mostrar algo encima de un marcador para así entender cómo debía estar estructurada la escena en Unity para que funcionara de forma correcta. Para esta prueba lo que se hizo fue atravesar el papel con un lápiz virtual, como se muestra en la figura 28.

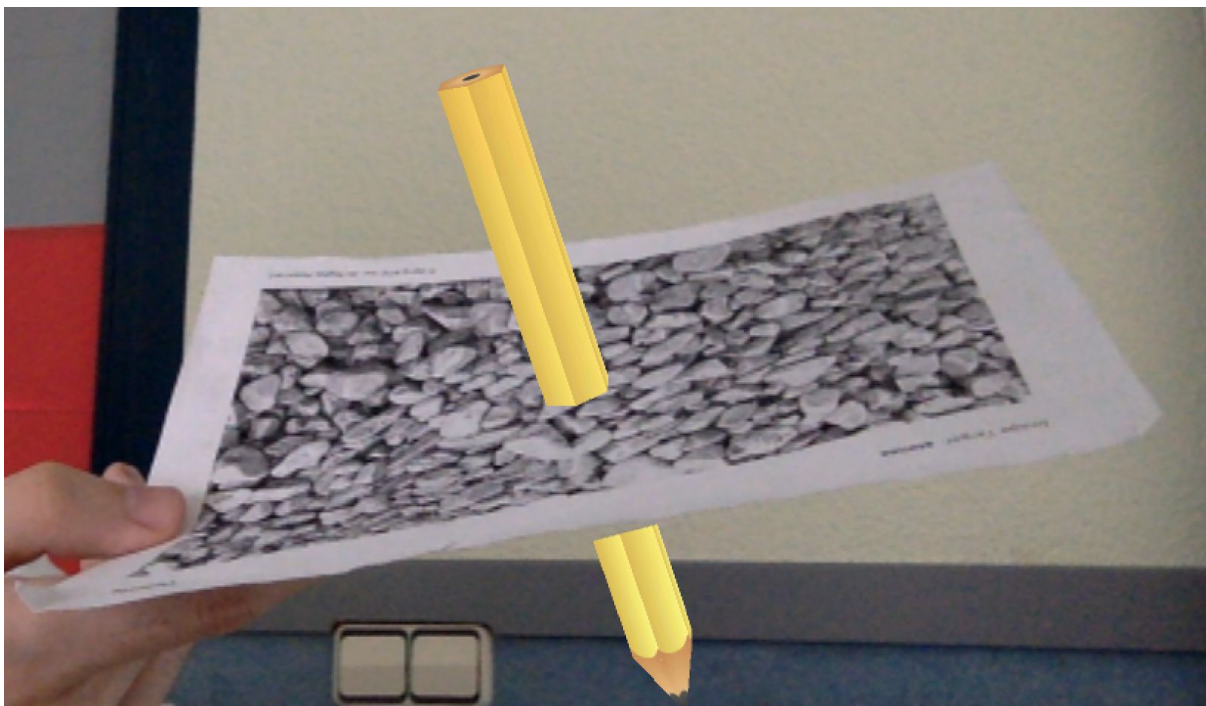


Figura 28. Lápiz virtual sobre marcador

De esta prueba se obtuvo diversa información. La más importante es que una vez detectada la imagen usada como marcador, los objetos virtuales que se habían preparado para aparecer sobre ella se escalan. Esto significa que si se utiliza un marcador el doble de grande, los objetos virtuales serán también el doble de grandes en comparación.

También se vio que, debido a que Vuforia detecta activamente dónde se encuentra la imagen, proporciona automáticamente oclusión sobre los objetos virtuales. Como se puede observar en la imagen, la parte del lápiz oculta por la imagen no aparece superpuesta sobre esta.

La siguiente prueba fue una prueba de estrés para comprobar qué tal funcionaba el seguimiento de varios marcadores a la vez. Para ello se usaron 5 marcadores distintos, cada uno con diseño obtenido de la figura 29.

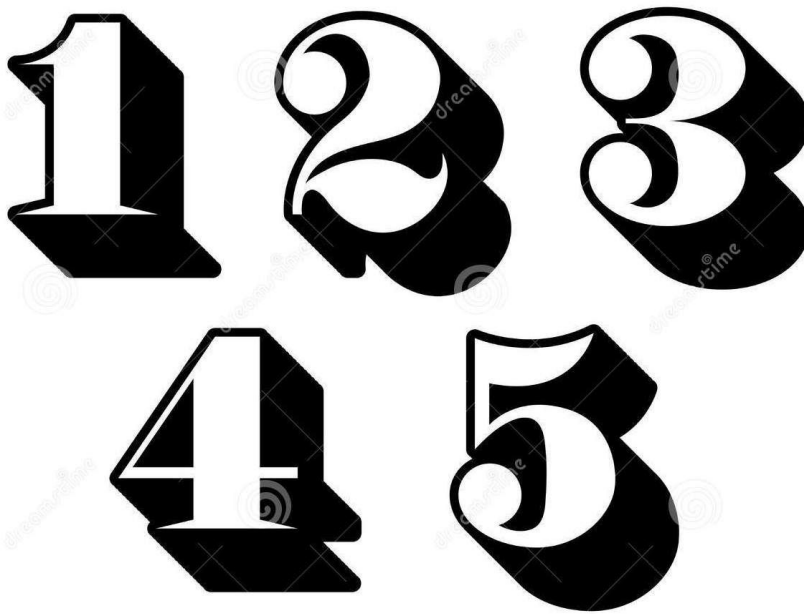


Figura 29. Imágenes de números utilizados como marcadores

Se creó una aplicación que hacía que cuando un marcador era detectado, le aparecía un *tick* encima para indicar que había sido reconocido. Este *tick* desaparecía cuando el marcador dejaba de ser detectado. Así, cumplía con una doble función: indicar si un marcador había sido detectado y mostrar el seguimiento de dicho marcador con el movimiento de la cámara.

Para facilitar más el seguimiento de dichos marcadores, se utilizó también una opción llamada *extended tracking* (seguimiento extendido) que permite utilizar los alrededores de cada marcador como referencia para, en caso de dejar de ser detectados, mantener los objetos virtuales creados por un marcador en el lugar donde ese marcador debería estar situado (suponiendo que el marcador esté en una posición fija).

El resultado final no fue suficientemente convincente como para implementar varios marcadores en el juego. Para poder detectar todos los marcadores había que acercar la cámara individualmente a cada uno lo suficiente como para que muchas veces se perdieran de vista los demás marcadores, que luego volvían a ser reconocidos más fácilmente al alejar la cámara debido al *extended tracking*. Además, la precisión y el seguimiento de estos marcadores se reducía cuantos más tenía detectados la aplicación.

La tercera prueba consistió en intentar aplicar la realidad aumentada a un proyecto ya existente. Para ello se utilizó la plataforma GREP (*Game Rules scEnario Platform* o plataforma de reglas y escenario de juego), que permite el diseño de escenarios en 3D y reglas de juego exportables luego en formato XML (*Extensible Markup Language* o lenguaje de marcas extensible) [19]. En concreto, se utilizó un escenario y unas reglas ya definidas, que simulaban una granja, y se implementó la

tecnología de Vuforia para, con la ayuda de GREP Player, proyectar dicha granja en un marcador definido, tal y como aparece en la figura 30.



Figura 30. Proyección de una granja de GREP Player sobre un marcador

Gracias a esta prueba, además de que por muchos elementos virtuales que se encontraran en la escena no se ve reducida ni empeorada la funcionalidad de la aplicación tanto como detectar varios marcadores, se comprobó que la gravedad no funcionaba de la forma natural a la que estamos naturalmente acostumbrados.

Después de varias pruebas más se vio que Vuforia permite manejar la gravedad de dos formas diferentes: la primera consiste en que uno de los marcadores actúe como la superficie sobre la que actúa la gravedad, de forma que esta se dirige perpendicularmente hacia el marcador. En este caso, todos los objetos virtuales son atraídos hacia el marcador (figura 31).

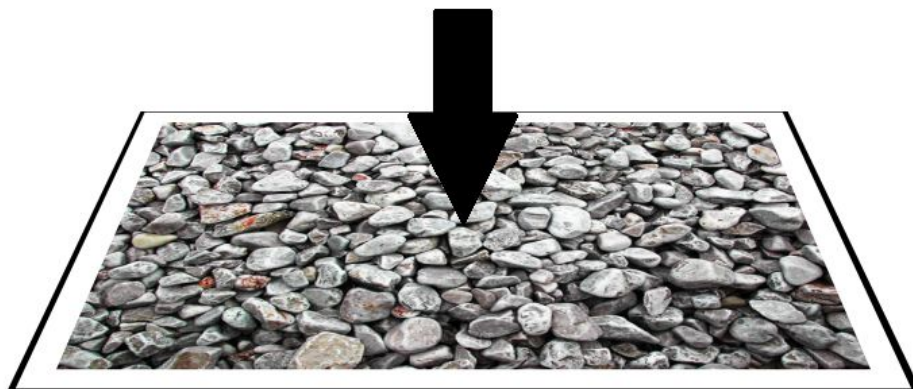


Figura 31. Representación de la gravedad virtual (flecha negra) con marcador horizontal

El problema con esta situación es que si el marcador no se encuentra paralelo al suelo, la gravedad no es la misma que sentimos nosotros (figura 32) y, por lo tanto, aunque el marcador se encuentre inclinado o volcado los objetos virtuales no se moverán de donde se encuentran.

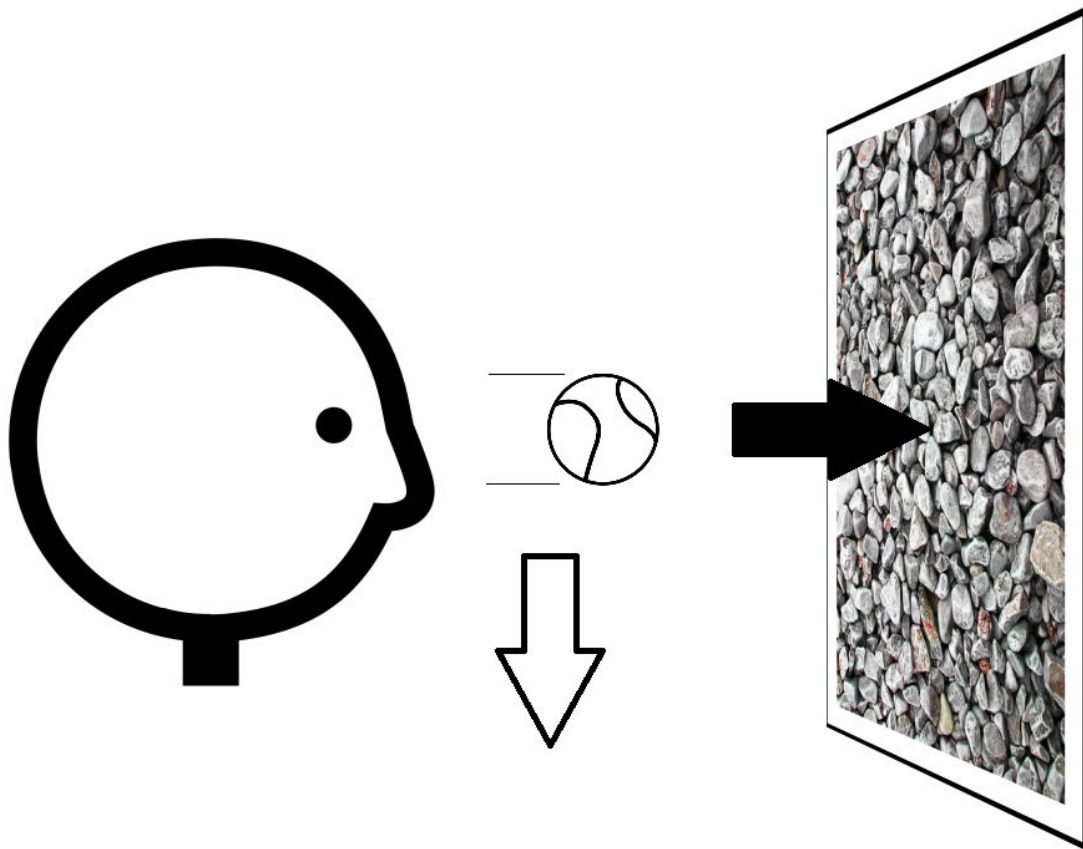


Figura 32. Representación de la gravedad virtual (flecha negra) contra la real (flecha blanca) con marcador no horizontal

La segunda forma de implementar la gravedad con Vuforia consiste en que esta siempre sea perpendicular hacia donde esté mirando la cámara, de forma que todos los objetos virtuales se dirigen siempre hacia la parte de abajo de la pantalla. Esta situación se muestra en la figura 33.

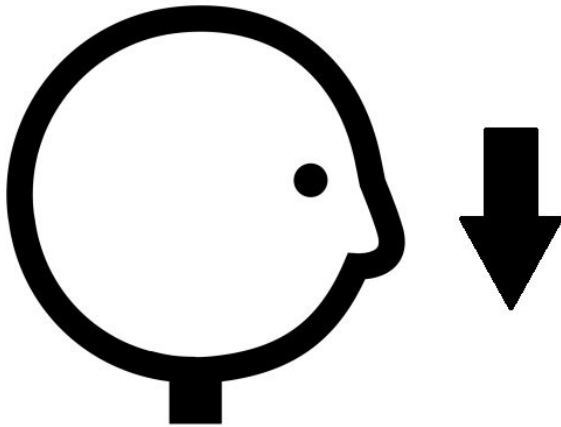


Figura 33. Representación de la gravedad virtual (flecha negra) con cámara vertical

En este caso, esto supone un problema: si la cámara se encuentra inclinada hacia delante, detrás o algún lado, los objetos virtuales no sentirán la misma gravedad que nosotros (figura 34). Esto significa que la gravedad solo será correcta cuando la cámara se encuentra perpendicular al suelo, sin ninguna clase de inclinación.

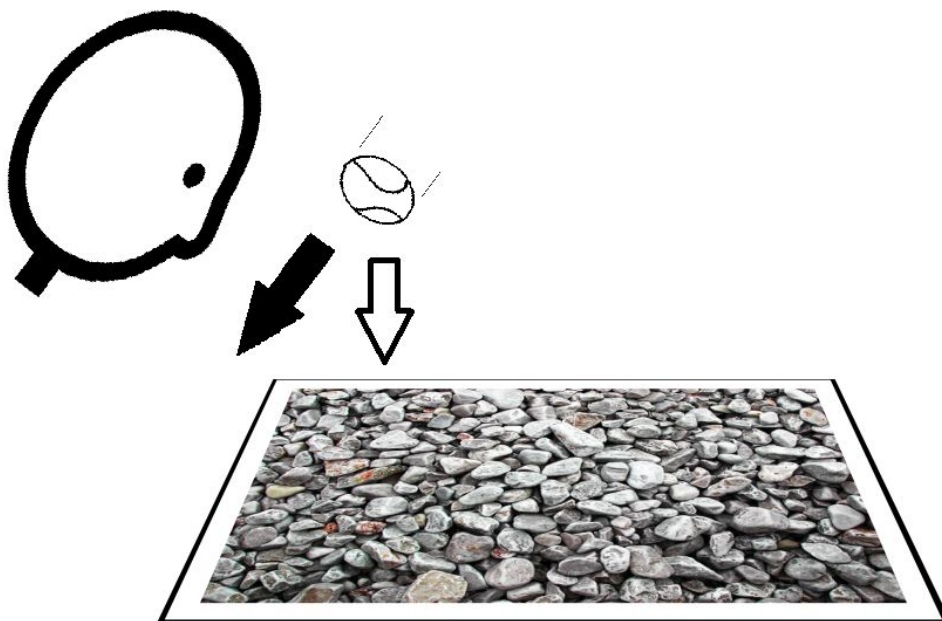


Figura 34. Representación de la gravedad virtual (flecha negra) contra la real (flecha blanca) con cámara no vertical

Este problema fue el que causó que se quisiera hacer uno de los dos juegos manejado con la misma gravedad que sentimos, sin importar la posición del marcador o del dispositivo.

Para solucionar el problema de la gravedad se probó una solución basada en dos marcadores: uno de ellos sirve como referencia para la gravedad, y en el segundo se desarrolla el juego, representado en la figura 35. De esta forma, mientras que el marcador usado para la gravedad se encuentre paralelo al suelo, los objetos virtuales situados sobre el marcador del juego seguirán la gravedad natural.

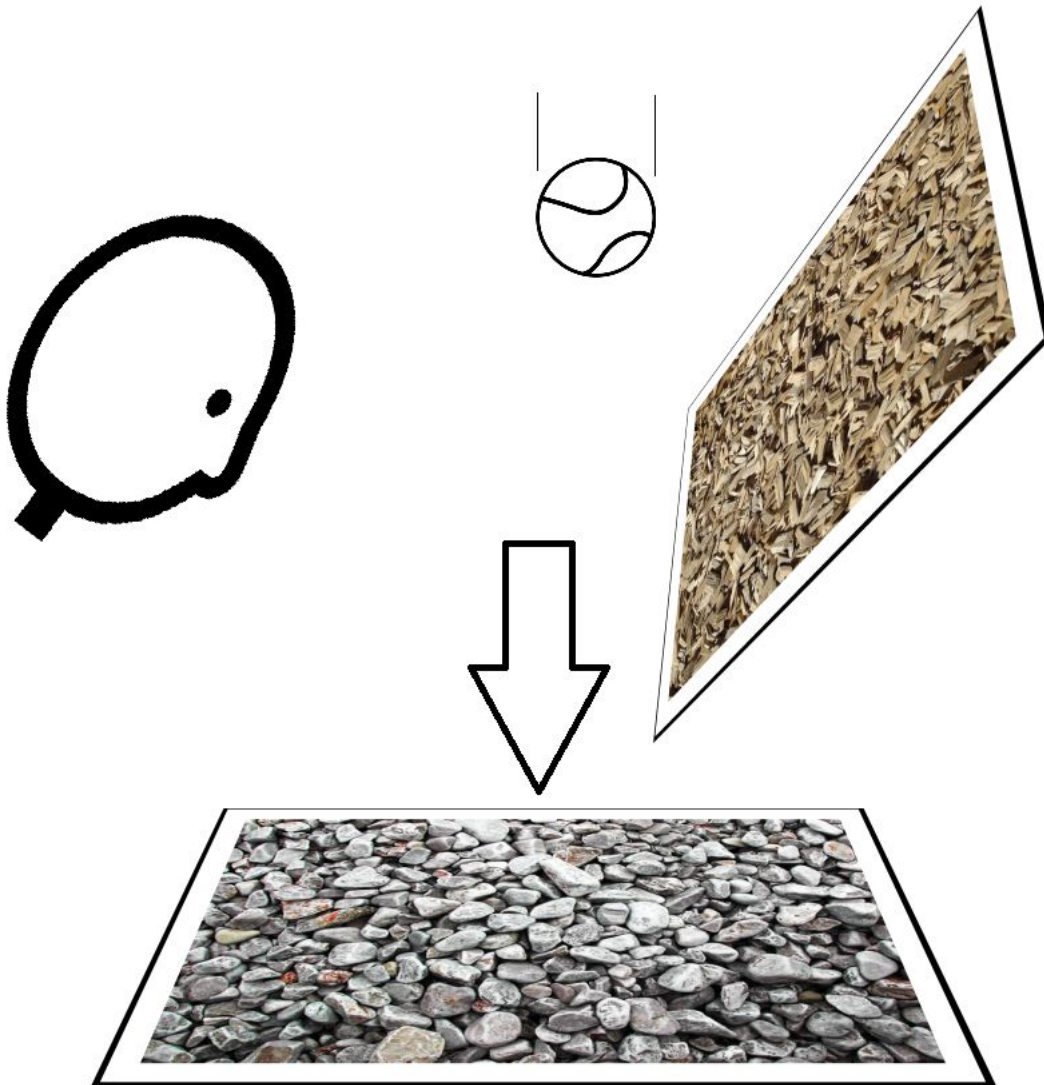


Figura 35. Representación de la gravedad real y virtual (flecha blanca) sin importar el ángulo de la cámara ni del marcador de juego (marcador con dibujos de madera)

Esta solución tenía varios fallos. El primero de ellos es que, al necesitar seguir varios marcadores a la vez, se reducía el funcionamiento de la aplicación. Además, en caso de que el marcador usado para controlar la gravedad desapareciera del campo visual, se perdía la gravedad por completo. Por otro lado, en caso de no

tener una superficie plana en la que colocar dicho marcador que estuviera alejada del suelo, la distancia a la que se encuentran los dos marcadores (suponiendo que el usuario esté de pie) es demasiado grande y el cálculo de la gravedad se vuelve impreciso.

Una vez descartada esta solución, se comprobó que si se cambiaba la orientación de la cámara en Unity, al probar el juego la gravedad era diferente (suponiendo que la opción de gravedad seleccionada sea la basada en la cámara). Es decir, si la cámara se sitúa con una orientación de 45° hacia abajo en Unity, a la hora de jugar y suponiendo que la cámara es perpendicular al suelo la gravedad ya no se dirige hacia abajo, sino que se va hacia abajo y alejándose del usuario (figura 36), con un ángulo de 45° .

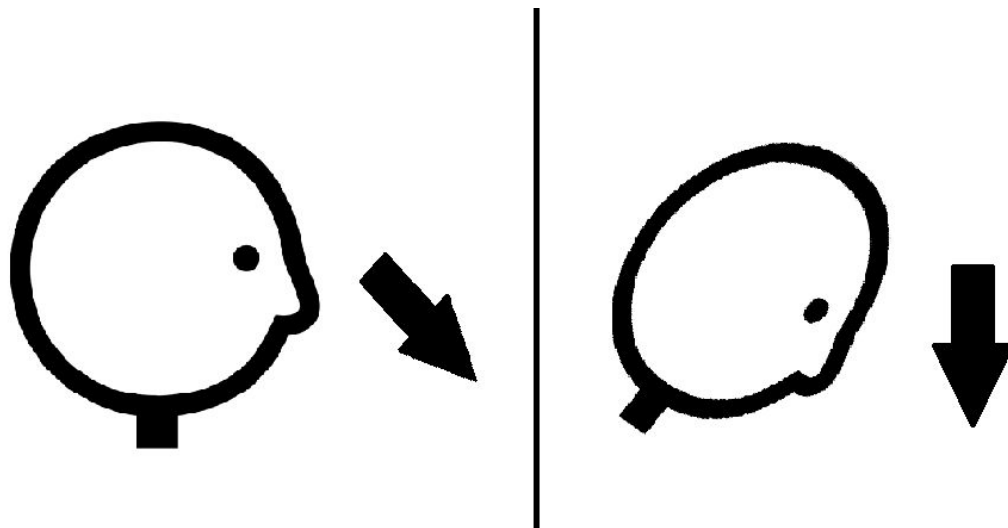


Figura 36. Representación de la gravedad virtual (flecha negra) dependiendo del ángulo de la cámara

Cuando se comprobó que esto funcionaba para todos los ángulos posibles, surgió la idea de obtener los ángulos en los que se encuentra posicionado el dispositivo y modificar en tiempo real los valores de la cámara de Unity, para así poder ajustar la gravedad a la que fuera correcta. Esta fue la solución que finalmente se adoptó.

En cuanto al menú de selección, para evitar que el usuario necesitara tocar la pantalla se hicieron unas pequeñas pruebas con los botones virtuales de Vuforia, que es lo que finalmente se utilizó. Estas pruebas se reutilizaron para crear los menús que se usaron en el juego.

Estas pruebas también sirvieron para comprobar que se podía asociar un botón virtual a un objeto que no fuera simple texto. Tras esto, se probó a mover el objeto trasladando con él el botón virtual, y como esta prueba resultó exitosa se procedió a implementar esto en el juego Arkanoid como el método de control.

5.2 Desarrollo de la aplicación

Para el desarrollo de la aplicación en Unity se dividió el juego en cuatro escenas diferentes: el menú de elección de juego, el menú de selección de dificultad y una escena para cada uno de los juegos.

La idea de cómo implementar la escena del menú principal surgió después de realizar las pruebas con los botones virtuales de Vuforia para el control del juego Arkanoid, y consta de diversos elementos: un plano que representa la imagen que la aplicación debe detectar, los botones virtuales con sus respectivos textos y la cámara virtual de Unity que permite detectar el dispositivo en el que se ejecuta la aplicación. Todos ellos son mostrados en la figura 37.

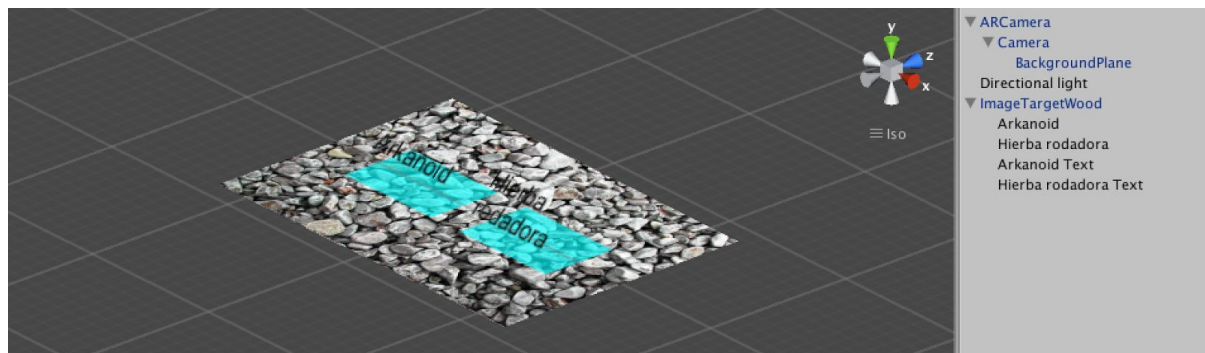


Figura 37. Objetos y jerarquía de la escena de selección de juego

Esta escena solamente utiliza una modificación de los *scripts* (guiones) utilizados por Vuforia para cambiar el comportamiento de los botones virtuales cuando se detecta su oclusión, por lo que todos los *scripts* están escritos en el lenguaje de programación C#.

La escena de selección de consiste básicamente en lo mismo que la escena anterior (figura 38), con la diferencia de que se recogen los valores elegidos en la elección de juego para redirigir a una escena o a otra una vez seleccionada la dificultad. Los *scripts*, al igual que la escena anterior, están escritos en el lenguaje C#.

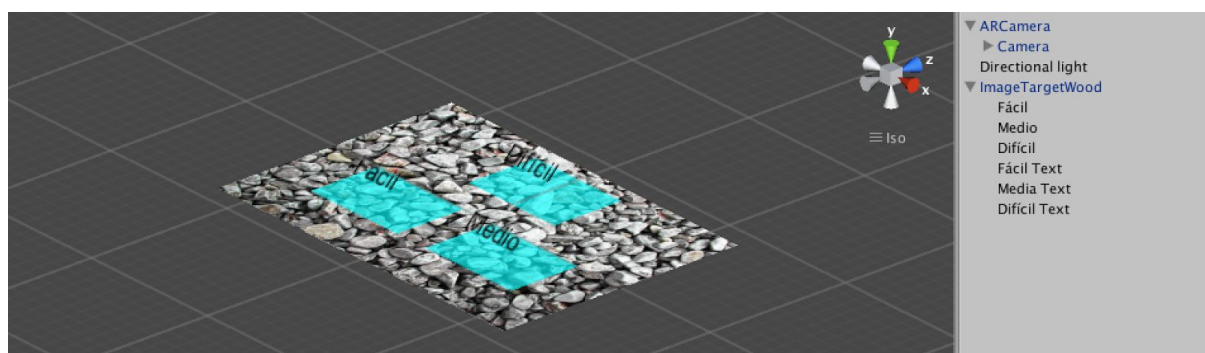


Figura 38. Objetos y jerarquía de la escena de selección de dificultad

Con respecto al juego Arkanoid, la comprobación de que los botones virtuales podían trasladarse junto con el objeto que los representa (la pelota, en este caso) dio lugar a la posibilidad de desarrollar el método de control a través del uso de la mano del jugador.

Esta escena cuenta con muchos más objetos (figura 39), dado que cada uno de los bloques que conforma el escenario es representado como un objeto adicional. Además, cada una de las paredes debe permanecer invisible para el usuario porque se superpondrían a la mano del usuario.

Los *scripts* necesarios para el funcionamiento de la aplicación, tanto para la mecánica del juego (como para cambiar el color de los bloques, destruir un bloque, poner en pausa el juego o comprobar el final de la partida) como el necesario para que se reconozca la oclusión de la pelota, han sido escritos en C# dado que este último también consiste en una modificación de los *scripts* originales de Vuforia. Al haber objetos en la escena que utilizan variables de otros objetos para comprobar la situación del juego actual, es más fácil el uso de variables compartidas a través de *scripts* que usan en el mismo lenguaje de programación (por ejemplo, si un *script* escrito en JavaScript utiliza una variable creada en un *script* escrito en C#, el *script* en C# debe ser compilado primero o la variable no estará disponible para el *script* en JavaScript).

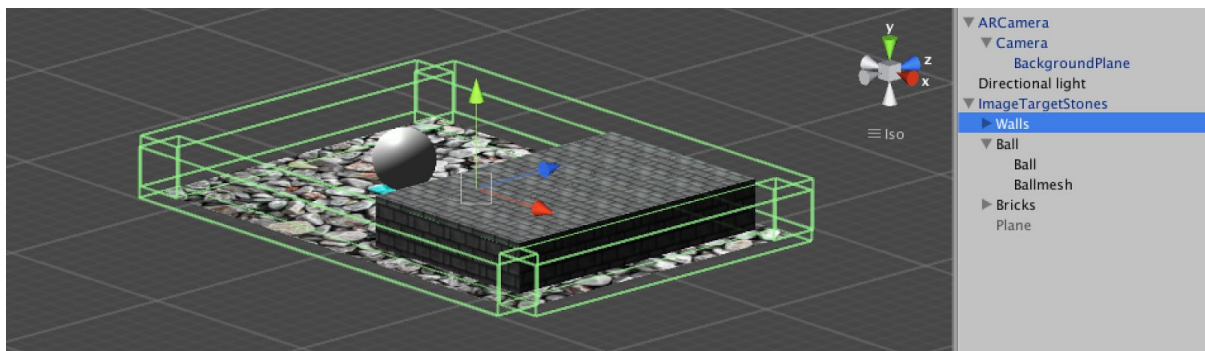


Figura 39. Objetos y jerarquía de la escena del juego Arkanoid

Por los mismos motivos que en la escena de Arkanoid, en la escena del juego Hierba Rodadora se escribieron los *scripts* en C# también. Entre los *scripts* utilizados se encuentran los que se encargan de modificar la perspectiva de la cámara de Unity para adaptar la gravedad, hacer que las cajas sean visibles o invisibles y desciendan si es su turno, sincronizar todas las cajas para indicar de cuál es el turno, poner en pausa el juego si el escenario desaparece del campo de visión, oscurecer los bloques que conforman el suelo para dar el efecto de sombra y comprobar el final de la partida.

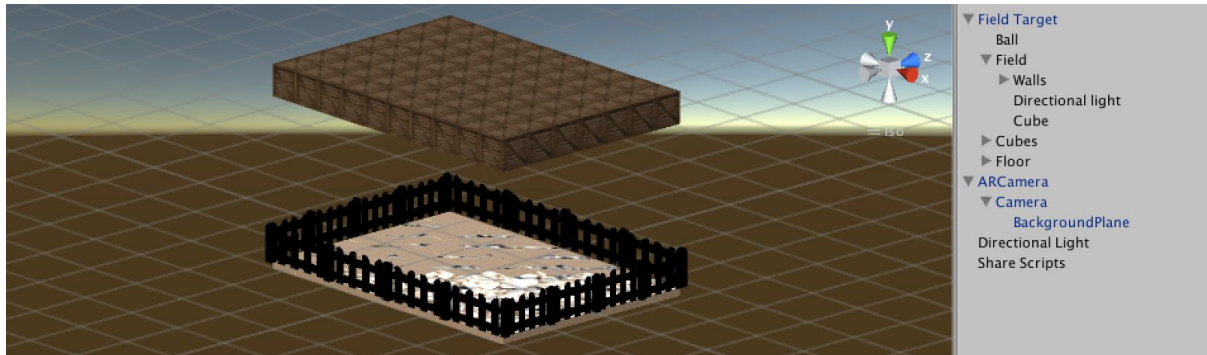


Figura 40. Objetos y jerarquía de la escena del juego Hierba Rodadora

En este caso, los objetos más importantes creados son las cajas que se encuentran suspendidas sobre el escenario, las paredes, la pelota y el suelo (figura 40), que está dividido en bloques para poder indicar dónde va a caer la siguiente caja por medio del oscurecimiento de estos bloques. Esto es así debido a que, con el cambio de perspectiva constante que sufre el tablero por el uso de la realidad aumentada, no es posible proyectar sombras virtuales de forma precisa sobre el tablero.

6. PRUEBAS

Una vez completado el desarrollo de la aplicación, se hicieron pruebas para comprobar que todas las funcionalidades estuvieran correctamente implementadas en ella.

En primer lugar, se probaron las tres dificultades de los dos juegos. Como se puede ver en las figuras 41 y 42, eligiendo diferentes dificultades se pueden apreciar cambios en los escenarios, además de otros cambios como la velocidad de la pelota o de las cajas que no se puede apreciar en las imágenes.

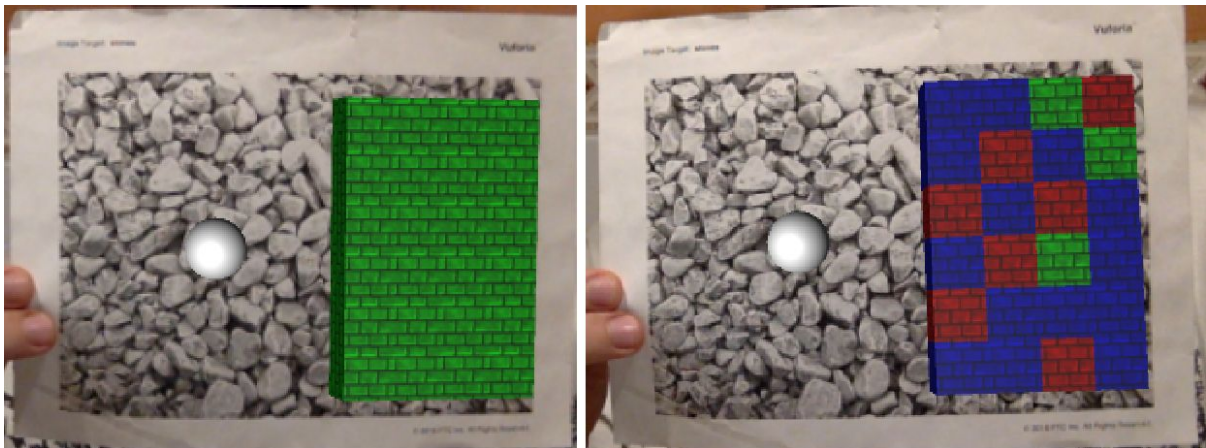


Figura 41. Juego de Arkanoid en nivel fácil (izquierda) y difícil (derecha)

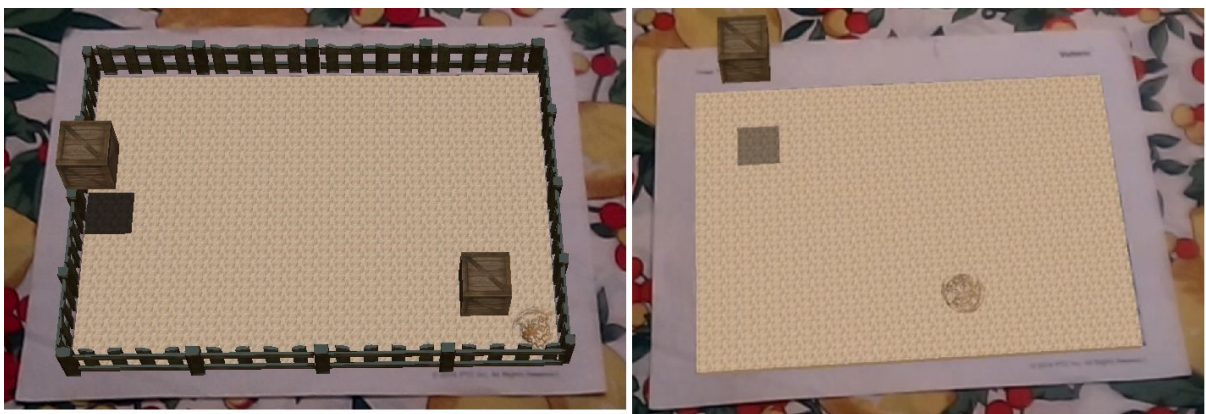


Figura 42. Juego Hierba Rodadora en nivel fácil (izquierda) y difícil (derecha)

Después, se comprobó que todas las opciones del menú principal eran fácilmente visibles y redirigían al menú correspondiente. También se observó que los textos fueran visibles desde una amplia variedad de ángulos de visión, como se muestra en la imagen 43.

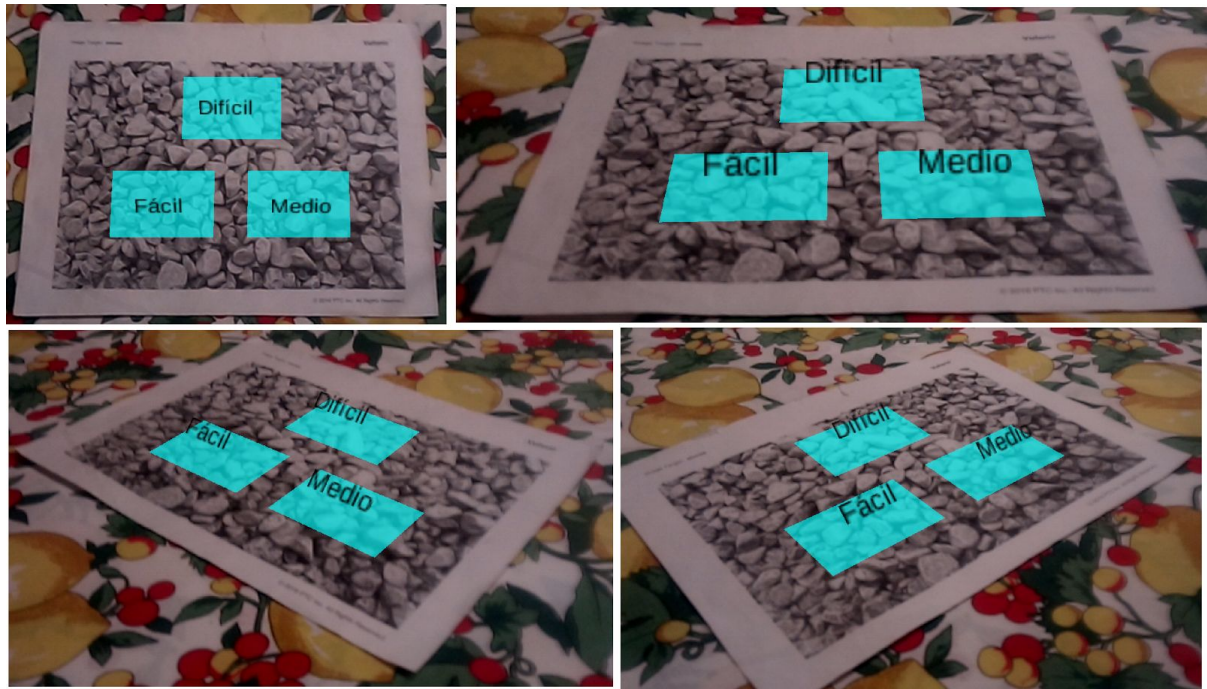


Figura 43. Menú de elección de dificultad desde distintos ángulos

Otro aspecto importante de los juegos era que pudieran ponerse en pausa con tan solo apartar la mirada del tablero. Después de comprobar esto con varios juegos, se pudo ver que ambos continuaban exactamente como habían sido dejados antes de perder el tablero del campo visual.

Por último, se comprobó que cuando se cumplían las condiciones que indican una victoria o una derrota en cada uno de los juegos, estos indicaban correctamente el fin de la partida (figura 44).



Figura 44. Imagen de victoria (izquierda) y derrota (derecha) en el juego Arkanoid

7. CONCLUSIONES

7.1 Objetivos cumplidos

Una vez el proyecto ha sido finalizado satisfactoriamente, se puede comprobar cuáles de los objetivos que habían sido planteados al principio del mismo han sido finalmente cumplidos:

El objetivo principal era crear un juego de realidad aumentada que se pudiera manejar completamente sin depender de elementos virtuales, a través de movimientos del mundo real en lugar de tocar la pantalla. Este objetivo ha sido totalmente cumplido, debido a que el menú de selección se controla pasando la mano por encima del marcador, al igual que el juego Arkanoid. Por otro lado, el juego Hierba Rodadora se maneja únicamente inclinando el marcador, sin necesidad de interactuar tampoco con el dispositivo.

El segundo objetivo era que la aplicación interfiriera lo menos posibles con otras actividades que el usuario pudiera estar realizando o que pudieran surgir. Esto se ha logrado de dos formas distintas: la primera es que los elementos virtuales creados ocupan lo mínimo posible del mundo real para así interferir lo menos posible con la visión del usuario.

La segunda forma de conseguir esto ha sido que, a pesar de necesitar un marcador para poder utilizar la aplicación, en caso de que se necesite prestar atención a otra cosa o que el usuario necesite desviar su mirada hacia otro lado, con que el marcador desaparezca del campo de visión del usuario la aplicación continuará tal y como la dejó, para poder cambiar su foco de atención rápidamente en caso de necesitarlo.

El último objetivo era desarrollar la aplicación tanto para gafas de realidad aumentada como para *smartphones*, lo cual ha sido cumplido. El desarrollo principal, así como las pruebas, se desarrollaron en un *smartphone* del que se portó el proyecto una vez finalizado a una versión para gafas de realidad aumentada, cambiando elementos como el ángulo en el que se ve la imagen (debido a que en las gafas de realidad aumentada se necesitan dos perspectivas ligeramente distintas para lograr el efecto tridimensional).

Como se ha podido comprobar, los tres objetivos principales han sido cumplidos con éxito. A pesar de que la planificación inicial del proyecto había sido de 4 meses y medio, como se puede comprobar en el apartado de gestión del proyecto, se ha extendido más de lo planeado. Esto también indica la dificultad de la planificación de un proyecto de *software*, ya que el tiempo estimado no incluía imprevistos que finalmente han retrasado el producto final durante meses.

7.2 Futuras líneas de trabajo

Este proyecto, por otro lado, ha dejado diversas líneas que se pueden mejorar en un futuro para ampliar la funcionalidad de esta aplicación o para mejorar aspectos ya existentes de la misma.

En primer lugar está la existencia del marcador. A pesar de que es una pieza clave de este proyecto, su necesidad hace que el usuario deba llevar el marcador encima para poder jugar. Una solución bastante sencilla sobre la que continuar esta aplicación puede consistir en permitir al usuario elegir la imagen con la que jugar, de forma que seleccione una imagen del dispositivo y la aplicación utilice dicha imagen como marcador. Otra opción es encontrar una forma de detectar la mano del usuario sin necesidad de un marcador, con la cual se podrían usar gestos de la misma para hacer aparecer el tablero de juego encima de la mano directamente.

Otro aspecto a mejorar, ligado con el punto anterior, es la inexistencia del tablero y las paredes en el juego del Arkanoid. Con una tecnología que permita detectar la mano del usuario, se puede conseguir superponer su silueta sobre los elementos virtuales para hacer la experiencia más realista.

Una línea más de desarrollo puede consistir en la adaptación de la aplicación a otras plataformas, como *smartphones* con sistema operativo iOS u otras gafas de realidad aumentada o virtual.

8. GESTIÓN DEL PROYECTO

A lo largo de este proyecto, se han utilizado varios recursos y se ha invertido tiempo de diverso personal involucrado. Para representar esto, se va a representar tanto el tiempo invertido en la creación de la aplicación y este informe como el coste del proyecto en un diagrama de Gantt y una tabla con los presupuestos, respectivamente.

Así mismo, también se va a exponer el método utilizado para la planificación del proyecto al desarrollar el software.

8.1 Planificación

La planificación del proyecto que se realizó al empezar el mismo se correspondía con el diagrama de Gantt de la figura 45.

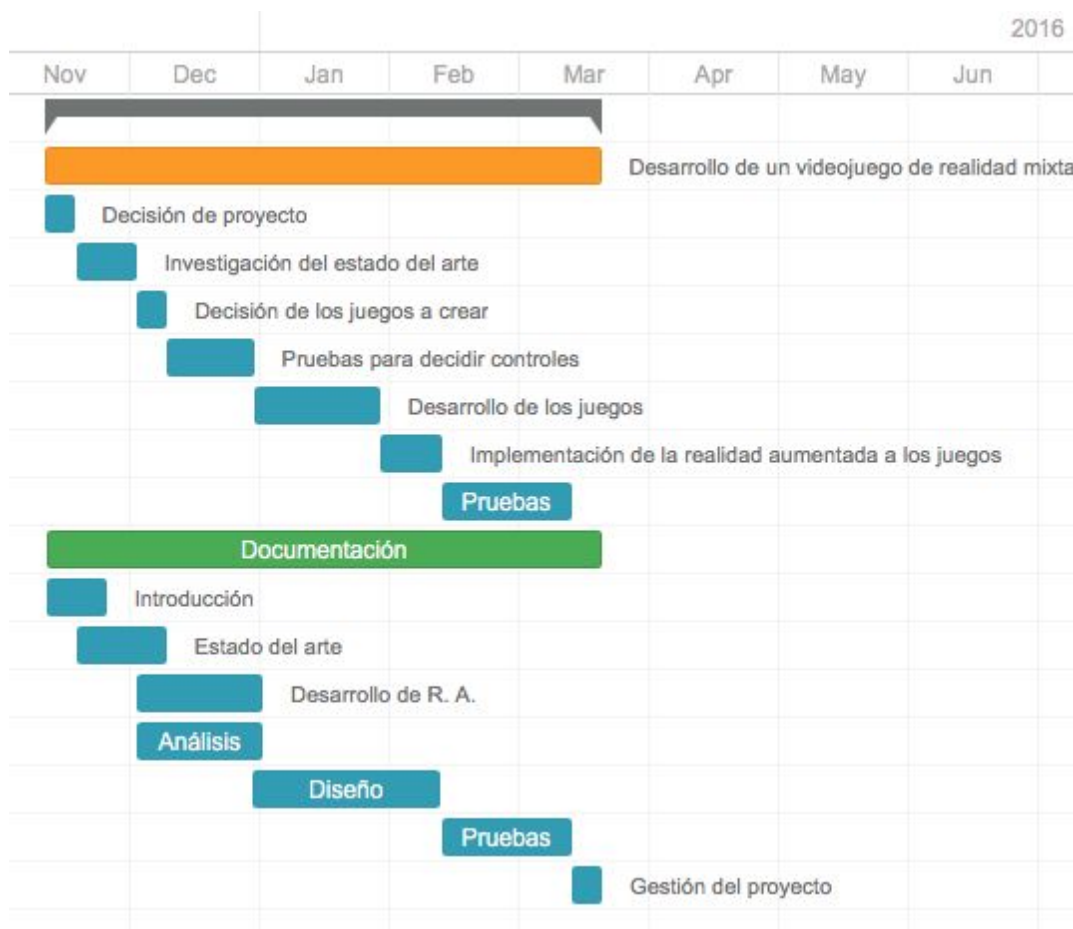


Figura 45. Diagrama de Gantt inicial del proyecto

Como se puede observar en el mismo, la estimación del fin del proyecto se situaba en la segunda mitad del mes de marzo de 2016. Sin embargo, debido a problemas que surgieron durante el desarrollo tanto de las pruebas para la tecnología de

Vuforia como del desarrollo de los juegos, además de la implementación de la realidad aumentada en ellos, el proyecto terminó por seguir el horario explicado en la figura 46.

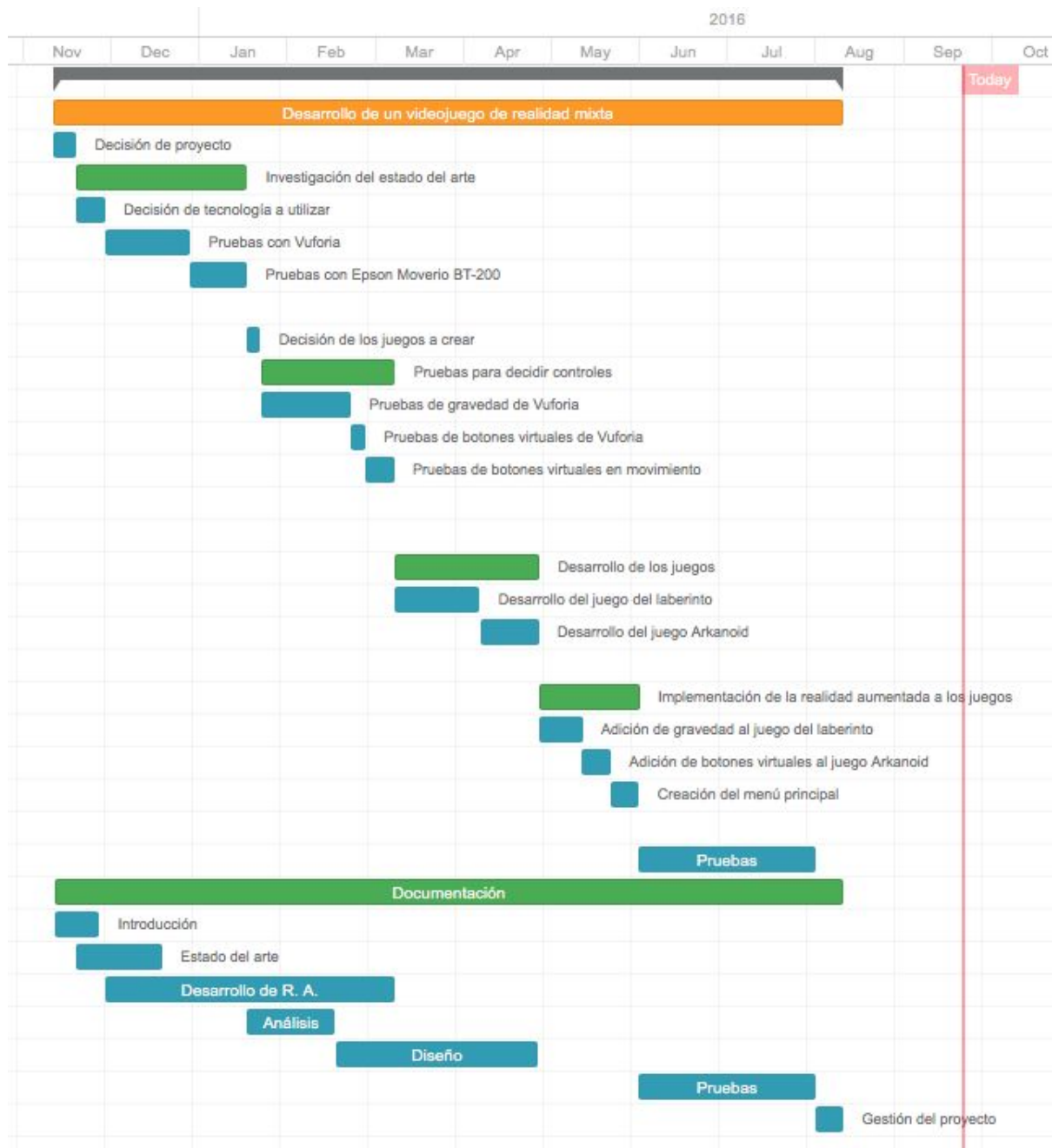


Figura 46. Diagrama de Gantt final del proyecto

Como se puede observar, además de contener más detalles y subtareas debido a lo poco definido que se encontraba el proyecto al principio, el resultado final ha sido que el proyecto fue acabado en la primera mitad de agosto de 2016, con un retraso de 4 meses y medio en comparación con lo inicialmente planeado.

8.2 Método de desarrollo

El método de desarrollo utilizado para este proyecto ha sido el denominado método en V, como se muestra en la figura 47.

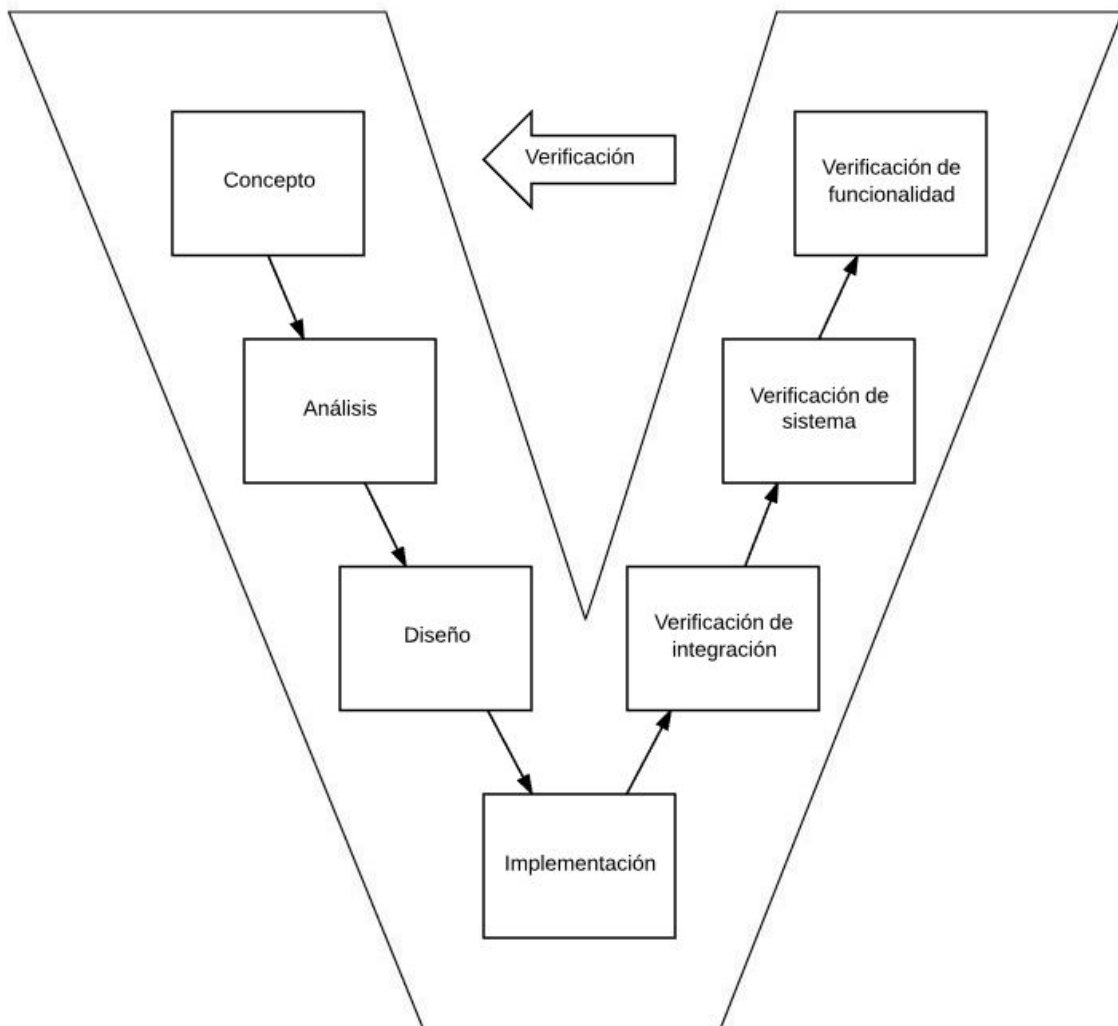


Figura 47. Método de desarrollo en V

Como se puede observar, la parte teórica del proyecto se sitúa en el lado izquierdo de la V, mientras que en el lado derecho están las tareas de validación y verificación. En cada una de estas tareas se comprueba que se cumpla lo especificado en su elemento correspondiente del lado izquierdo de la V, permitiendo así tener un *feedback* (retroalimentación) en cada una de las fases que permite modificar la implementación en caso de ser necesario.

8.3 Presupuesto

Para realizar el presupuesto de este proyecto se han tenido en cuenta dos factores: el sueldo de las personas involucradas en el mismo y el coste tanto del *software* como del hardware utilizado.

En primer lugar, el número de horas trabajadas, asumiendo una jornada de 4 horas de trabajo diarias durante días hábiles por parte del alumno, que ha ejercido diversos papeles a lo largo del proyecto. Entre ellos se encuentran el de cliente, analista, diseñador, desarrollador, implementador y *tester* (ensayador), cuyos salarios han sido obtenidos del Boletín Oficial del Estado. Los resultados se pueden observar en la tabla 57.

Papel	Tiempo trabajado (horas)	Salario (€/hora)	Coste total (€)
Analista	80	11,8	944
Diseñador	80	16	1280
Desarrollador	140	7,6	1064
Implementador	100	5,3	530
Tester	140	4,8	672
Total			4490

Tabla 57. Coste de personal

Por otro lado, el coste del hardware y el software utilizado (asumiendo el uso de una licencia de Vuforia para eliminar la marca de agua) es el indicado en la tabla 58.

Recurso	Precio (€)	Vida útil (meses)	Uso estimado (meses)	Coste total (€)
MacBook Pro	1300	72	8	145
Elephone P900	240	56	8	35
Epson Moverio BT-200	700	56	8	100
Vuforia	500	-	-	500
Total				780

Tabla 58. Coste de hardware y software

Así pues, teniendo en cuenta los costes totales tanto de personal como de recursos, el coste total del proyecto asciende a la cantidad de 5720 €.

9. BIBLIOGRAFÍA

1. Fernández, Samuel. "España, territorio smartphone." *Xataka Móvil*. Weblogs SL, 30 de enero de 2016.
<http://www.xatakamovil.com/movil-y-sociedad/espana-territorio-smartphone>.
Accedido el 26 de septiembre de 2016.
2. Marr, Bernard. "15 Noteworthy Facts About Wearables In 2016." *Forbes*. Forbes Media, 18 de marzo de 2016.
<http://www.forbes.com/sites/bernardmarr/2016/03/18/15-mind-boggling-facts-about-wearables-in-2016>. Accedido el 26 de septiembre de 2016.
3. "Real-world." *Your Dictionary*. LoveToKnow Corporation.
<http://www.yourdictionary.com/real-world>. Accedido el 26 de septiembre de 2016.
4. "What is a Virtual World?" *Techopedia*. Janalta Interactive.
<https://www.techopedia.com/definition/25604/virtual-world>. Accedido el 26 de septiembre de 2016.
5. Milgram, Paul, et al. "Augmented reality: A class of displays on the reality-virtuality continuum." *Photonics for industrial applications*. International Society for Optics and Photonics, 1995.
6. Flórez Aristizábal, Leandro. "Realidad aumentada y realidad mixta." *ResearchGate*. ResearchGate, mayo de 2016.
https://www.researchgate.net/publication/236879222_REALIDAD_AUMENTADA_Y_REALIDAD_MIXTA. Accedido el 26 de septiembre de 2016.
7. Cassella, Dena. "What is Augmented Reality (AR): Augmented Reality Defined, iPhone Augmented Reality Apps and Games and More." *Digital Trends*. Digital Trends, 3 de noviembre de 2009.
<http://www.digitaltrends.com/features/what-is-augmented-reality-iphone-apps-games-flash-yelp-android-ar-software-and-more/>. Accedido el 26 de septiembre de 2016.
8. Bejerano, Pablo. "El origen de la realidad aumentada." *Blogthinkbig.com*. Telefónica, 7 de agosto de 2014.
<http://blogthinkbig.com/realidad-aumentada-origen/>. Accedido el 26 de septiembre de 2016.
9. Azuma, Ronald T. "A survey of augmented reality." *Presence: Teleoperators and virtual environments* 6.4 (1997): 355-385.
10. "Realidad aumentada: la historia desde la ciencia ficción hasta la aplicación práctica." *Club Xataka*. Weblogs SL, 4 de junio de 2016.
<http://www.xataka.com/club/realidad-aumentada-la-historia-desde-la-ciencia-ficcion-hasta-la-aplicacion-practica>. Accedido el 26 de septiembre de 2016.

11. Galer, Susan. "7 Ways To Benefit From Augmented Reality Devices." *Digitalist Magazine*. SAP, 13 de febrero de 2014.
<http://www.digitalistmag.com/technologies/mobile-applications/2014/02/13/7-ways-to-benefit-business-01244412>. Accedido el 26 de septiembre de 2016.
12. Larkin, Francesca. "Advantages and disadvantages of Augmented Reality." *Behind the Spin*. PRCA, 3 de mayo de 2011.
<http://www.behindthespin.com/features/advantages-disadvantages-of-augmented-reality>. Accedido el 26 de septiembre de 2016.
13. "Ventajas y desventajas de la realidad aumentada." *Gigatecno*. Blogger, septiembre de 2014.
<http://gigatecno.blogspot.com.es/2014/09/ventajas-y-desventajas-de-la-realidad.html>. Accedido el 26 de septiembre de 2016.
14. Azuma, Ronald, et al. "Recent advances in augmented reality." *IEEE computer graphics and applications* 21.6 (2001): 34-47.
15. "Global mobile OS market share in sales to end users from 1st quarter 2009 to 1st quarter 2016." *Statista*. Statista, agosto de 2016.
<https://www.statista.com/statistics/266136/global-market-share-held-by-smart-phone-operating-systems/>. Accedido el 26 de septiembre de 2016.
16. Arth, Clemens, et al. "The history of mobile augmented reality." *arXiv preprint arXiv:1505.01319* (2015).
17. Amin, Dhiraj y Govilkar, Sharvari. "Comparative Study of Augmented Reality SDK's." *International Journal on Computational Sciences & Applications (IJCSA)* Vol. 5, No.1, febrero 2015
18. "Attributes of an Ideal Image Target." *Vuforia Developer Library*. Vuforia.
https://library.vuforia.com/articles/Best_Practices/Attributes-of-an-Ideal-Image-Target. Accedido el 26 de septiembre de 2016.
19. Zarraonandia, Telmo, Paloma Diaz, and Ignacio Aedo. "Using combinatorial creativity to support end-user design of digital games." *Multimedia Tools and Applications* (2016): 1-26.

ANEXO I: MANUAL DE USUARIO

Estas instrucciones están pensadas para usuarios de las gafas Epson Moverio BT-200. En caso de utilizar un *smartphone* como dispositivo, el mundo real se verá a través de la pantalla del dispositivo.

Nada más iniciar la aplicación, la pantalla del dispositivo no mostrará nada, de forma que todo el campo visual del usuario estará libre. Para iniciar el juego, el usuario debe poner el marcador en el campo visual de las gafas dirigiendo la mirada hacia él. Cuando se encuentre en el campo de visión, los elementos mostrados en la figura 48 aparecerán encima del marcador.

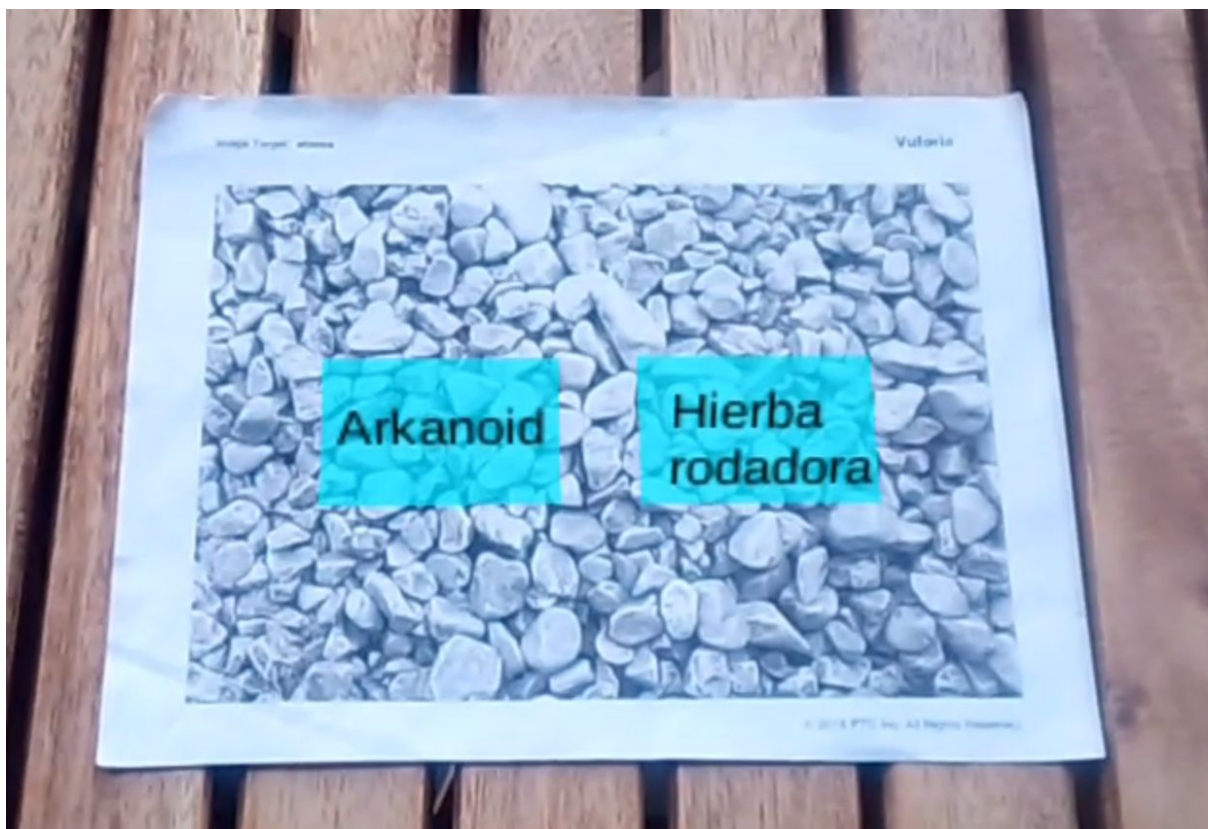


Figura 48. Menú de selección de juego

Para seleccionar el juego al que se desea jugar, el usuario debe posicionar la mano encima del título del juego correspondiente, como se indica en la figura 49.

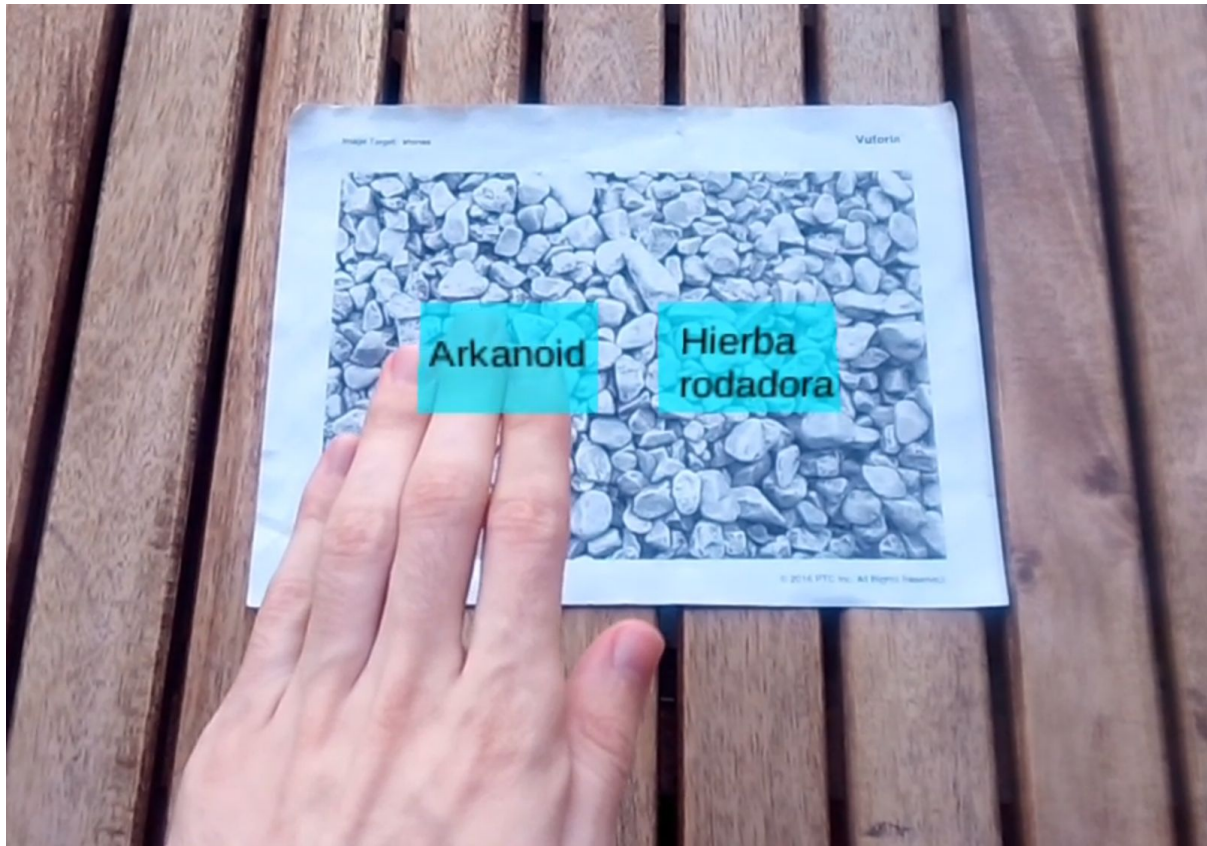


Figura 49. Modo de selección de juego

Para seleccionar el juego adecuado, a continuación se incluye la descripción de cada uno de los juegos:

- **Arkanoid:** En este juego, una pelota va a empezar a aparecer rebotando en unos ladrillos de la parte superior del tablero. El objetivo es que la pelota, a base de rebotar en dichos ladrillos, destruya todos ellos.

Para lograr esto, la pelota no debe llegar a la parte de abajo del tablero. En lugar de mover una pequeña plataforma a través de botones como en el juego original, la pelota rebotará hacia la parte superior del tablero cuando sea "tocada" con la mano.

- **Hierba Rodadora:** En este juego, unas cajas van a empezar a caer del cielo a una velocidad fija. El objetivo es mover la hierba rodadora para esquivar dichas cajas, hasta acabar en el último punto posible del tablero que no quede ocupado por ellas.

Para lograr esto, la hierba se moverá usando la gravedad del mundo real: inclinando la hoja se conseguirá que la hierba ruede hacia el punto más bajo del tablero. Esta gravedad, sin embargo, no permite hacer movimientos como saltos al mover súbitamente el tablero hacia arriba: la única fuerza que actúa activamente sobre la bola es la gravedad.

Una vez elegido el juego, y siempre que el marcador siga en el campo de visión del usuario, las opciones disponibles cambiarán de los títulos de los juegos a tres posibles dificultades: fácil, medio y difícil, como se muestra en la figura 50.

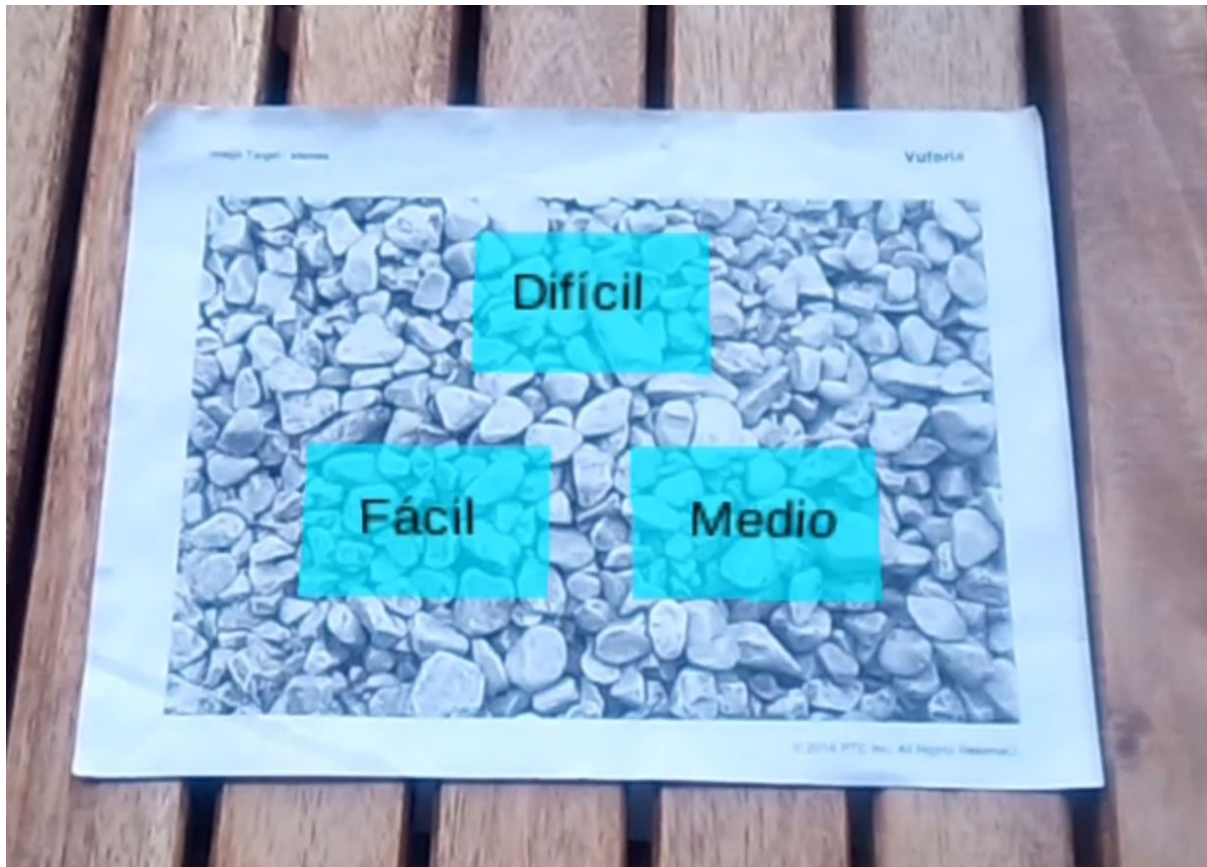


Figura 50. Menú de selección de dificultad

Para seleccionar la dificultad adecuada, a continuación se incluyen las condiciones de cada uno de los juegos:

- **Arkanoid:** El juego consta de 3 niveles de dificultad:
 - **Fácil:** todas las cajas necesitarán solamente 1 golpe para desaparecer, la pelota va a una velocidad lenta.
 - **Medio:** algunas cajas necesitarán 2 golpes para desaparecer, la pelota va a una velocidad moderada.
 - **Difícil:** algunas cajas necesitarán 2 o 3 golpes para desaparecer, la pelota va a una velocidad rápida.
- **Hierba Rodadora:** El juego consta de 3 niveles de dificultad:
 - **Fácil:** las cajas caen cada 2.5 segundos, las vallas hacen más difícil que la hierba rodadora caiga del tablero.
 - **Medio:** las cajas caen cada 1.75 segundos.

- **Difícil:** las cajas caen cada segundo, no hay vallas que frenen la caída del tablero.

En el caso de haber seleccionado en juego Arkanoid, en la pantalla aparecerá el escenario de la figura 51 encima del marcador.

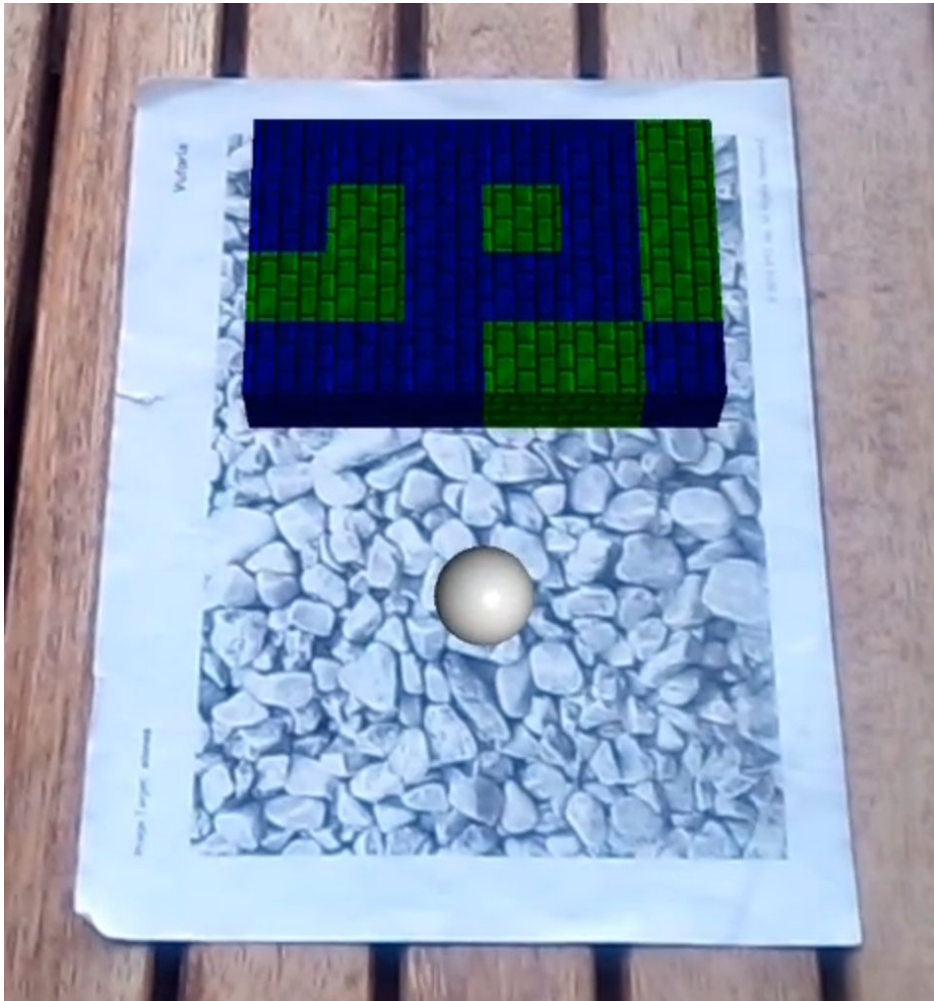


Figura 51. Escenario inicial del juego Arkanoid

Los controles de este juego son los siguientes:

- Hacer que la pelota rebote hacia arriba: tocar la pelota que aparece en la cámara con la mano en el mundo real (figura 52).
- Pausar el juego: basta con perder de vista el tablero para que el juego quede suspendido hasta que vuelva a estar a la vista.
- Volver al menú de selección: no tocar la pelota para perder la partida devuelve automáticamente al menú inicial.

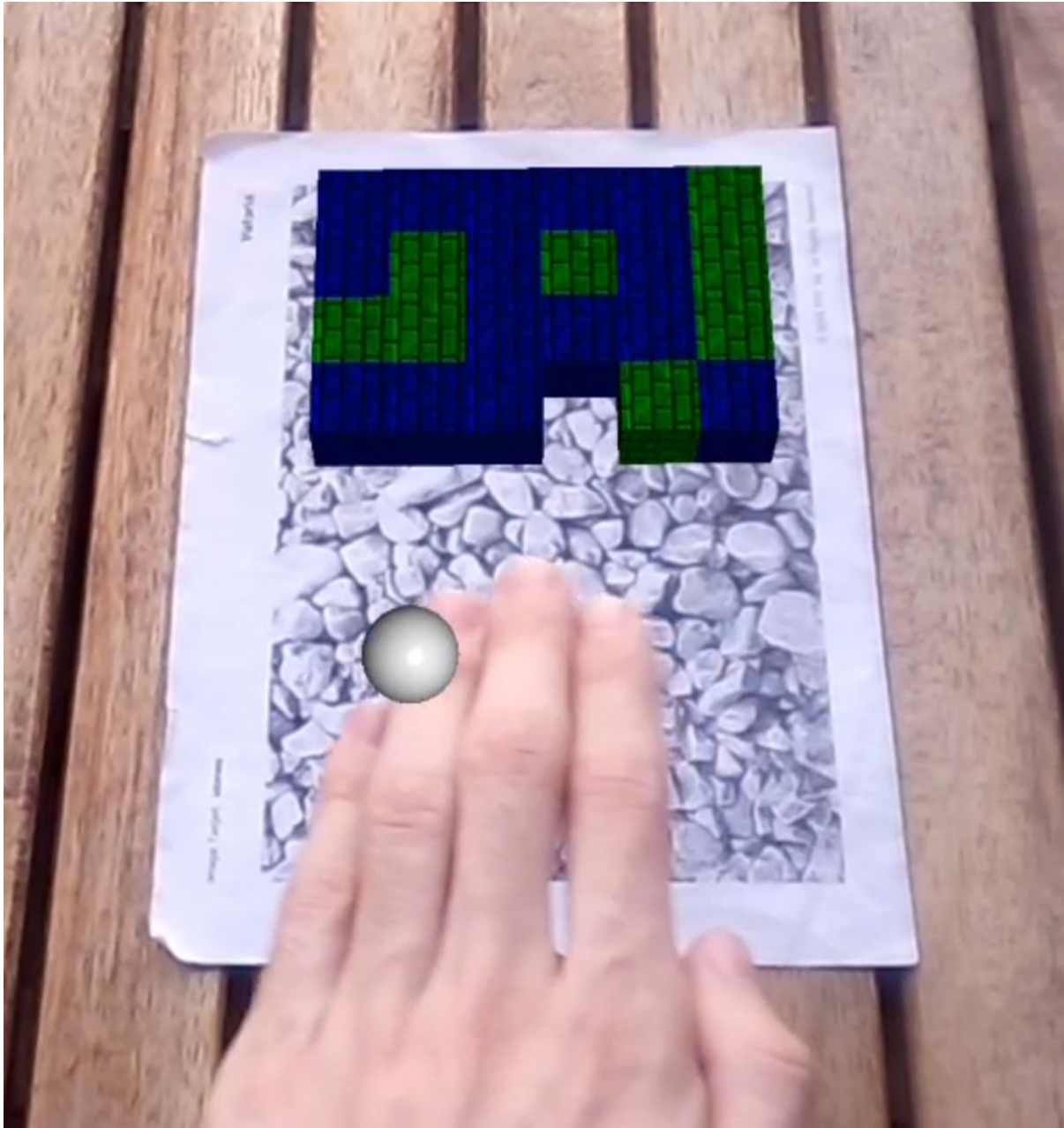


Figura 52. Modo de juego de Arkanoid

Si el usuario consigue eliminar todos los bloques, el juego pasará automáticamente al siguiente nivel de dificultad. En caso de que la pelota llegue a la parte inferior de la pantalla y el usuario pierda la partida, volverá automáticamente al menú de selección de juego.

En el caso de haber seleccionado en juego Hierba Rodadora, en la pantalla aparecerá el escenario de la figura 53 encima del marcador.

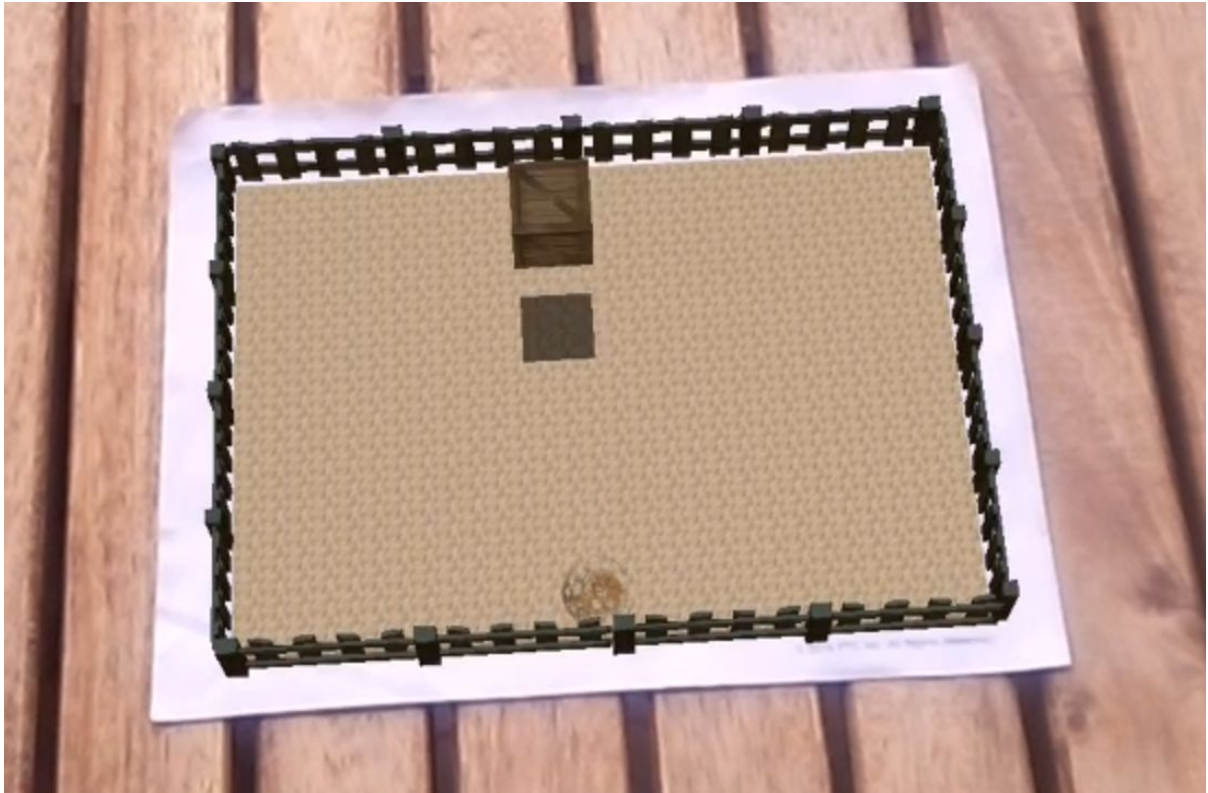


Figura 53. Escenario inicial del juego Hierba Rodadora

Los controles de este juego son los siguientes:

- Mover la hierba rodadora: inclinar el tablero hacia la zona a la que se quiera llevar la hierba (figura 54).
- Pausar el juego: basta con perder de vista el tablero para que el juego quede suspendido hasta que vuelva a estar a la vista.
- Volver al menú de selección: volcar el tablero para perder la partida devuelve automáticamente al menú inicial.

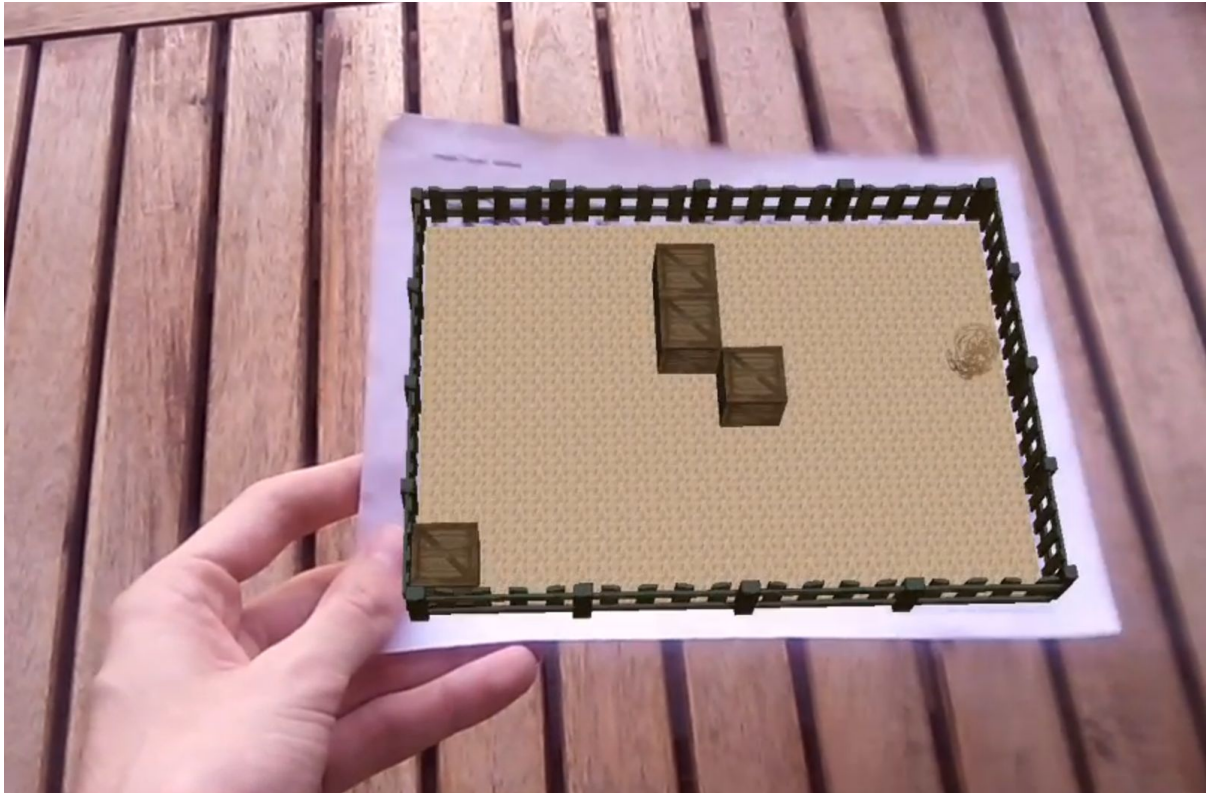


Figura 54. Modo de juego de Hierba Rodadora

Si el usuario consigue que la pelota siga en el tablero sin ser aplastada cuando caigan todas las cajas, el juego pasará automáticamente al siguiente nivel de dificultad. En caso de que la pelota caiga del tablero o sea aplastada por una caja y el usuario pierda la partida, volverá automáticamente al menú de selección de juego.

ANEXO II: GLOSARIO

- **Smartphone:** teléfono inteligente.
- **PDA:** Personal Digital Assistant o asistente digital personal.
- **App:** aplicación.
- **Wearable technology:** tecnología ponible.
- **Software:** componentes lógicos de un sistema informático.
- **GPU:** Graphical Processing Unit o unidad de procesamiento gráfico.
- **RAM:** Random-Access Memory o memoria de acceso aleatorio.
- **SDK:** Software Development Kit o kit de desarrollo de software.
- **API:** Application Programming Interface o interfaz de programación de aplicaciones.
- **Hardware:** componentes físicos de un sistema informático.
- **GPS:** Global Positioning System o sistema de posicionamiento global.
- **HMD:** Head-Mounted Display o gafas de realidad virtual.
- **KARMA:** Knowledge-based Augmented Reality for Maintenance Assistance o realidad aumentada basada en el conocimiento para la asistencia en el mantenimiento.
- **HUD:** Head-Up Display o visualización cabeza-arriba.
- **ARAS:** Augmented Reality-Assisted Surgery o cirugía asistida por realidad aumentada.
- **PSP:** PlayStation Portable.
- **On-line:** en línea.
- **Smartwatch:** reloj inteligente.
- **PS Vita:** PlayStation Vita.
- **QCAR:** Qualcomm Augmented Reality o realidad aumentada de Qualcomm.
- **PTC:** Parametric Technology Corporation o corporación de tecnología paramétrica.
- **2D:** dos dimensiones.
- **3D:** tres dimensiones.

- **Smart terrain:** terreno inteligente.
- **Script:** guión.
- **Feedback:** retroalimentación.
- **Tester:** ensayador.

DEVELOPMENT OF A MIXED-REALITY VIDEO GAME

Introduction

In the last few years, technology has suffered one of the most important advances of recent history: the so-called smartphones. These devices have managed to take concepts such as applications or the Internet to a much wider public than previously created devices that could also make use of them, like laptops or PDAs (Personal Digital Assistant).

As of January, 2016, approximately 80% of the adult population is already using a smartphone [1]. The fact that buying one of these devices is more and more accessible to the general public and the network value generated because of the extensive use of them for communicating, thus making it necessary to have one in order to be able to communicate with others, make these numbers keep rising more and more.

However, communications aren't the only feature of smartphones: since their creation, the so-called applications or apps have allowed to take utilities and games anywhere the smartphone can be taken, becoming this way in a revolution for the video games world, too. The popularity of apps has allowed users to spend their time playing on their small devices, without the need of buying a bigger and more expensive device used exclusively for this purpose, such as video consoles or PCs (Personal Computer).

On the other hand, this decade has also seen the rising of a technology directly related to smartphones: the so-called wearable technology, among which the smartwatches and augmented reality glasses must be noted. Although these technologies have not had the same reception as smartphones due to problems like software limitations, battery duration or their price, the number of users of these technologies has kept growing and it is foreseen a bigger growth in the next few years [2].

In the case of augmented reality glasses, specifically, the problem that has caused them not to be as extended is the high price of the available devices in the market, which makes the average user not even think about buying them.

However, this does not mean that in a not so distant future they will not have an utility in people's everyday lives, and with that thought in mind has been created this project, that tries to demonstrate that with a technology not so distant from the one that we currently have we can achieve objectives that will put it to better use when the reach of the technology grows larger.

Objectives

The main objectives of this project are the following ones:

- **Development of an augmented reality game for augmented reality glasses:** the main objective of this project is the development of a mixed reality game, more specifically an augmented reality game, that provides the user with a deeper immersion than the one that can be provided through the small screen of a smartphone. Creating it for augmented reality glasses can manage that, but this also makes it necessary that the application can be controlled without the need of any virtual element. This means that every control must be through the use of typical movements of the real world, such as touching or tilting, instead of touching a screen since the hands of the user must be free for a deeper immersion.
- **Minimal interference with the real world:** another of the objectives of this project is that said application tries to interfere as little as possible with other activities that can be happening while the game is running or that can suddenly interrupt the game. Given that the application is thought to be used anytime, anywhere, if it required additional equipment other than the device this objective could not be fulfilled.
- **Portability to smartphones:** the third and last objective consists in that, even though the application is mainly developed for its use with augmented reality glasses, it must be possible to use it with a smartphone too. The reason of this objective is that although playing a game with a smartphone in the hands of the user reduces the playability of the application (while by using augmented reality glasses you have more real world elements available to provide a better control and a wider field of view) a very high percentage of the population owns and uses smartphones daily. To make it available to a wider public and to be able to take clearer captures of scenarios of the game that, otherwise, would not be seen as clearly because the image must be duplicated in augmented reality glasses to produce a sensation of depth, the application will also be developed for smartphones.

Games election

In order to create a simple application that allowed to explore diverse aspects of augmented reality, it was decided that the games that would be created in this project would be adaptations of classic video games.

The first video game that will be adapted to augmented reality is the widely known Breakout or Arkanoid. Created in 1970 by the American company Atari (and later in 1986 its most popular version, called Arkanoid, was created by the company Taito), Breakout is a game where the user must control a paddle that makes the ball rebound whenever it touches it, trying that the ball does not pass to the lower part of the stage, under the paddle. Each level is completed whenever the ball destroys all

the bricks located in the upper part of the stage by rebounding against them. An example of a scenario of this game is shown in figure 17.



Figure 17. Scene from the game Breakout

The reason why this game was chosen to be adapted to augmented reality is because of the simplicity of its game mechanics, that allows to focus on finding a more innovative control mode for this well-known classic.

The second game, in order to not only make adaptations of existing games, is an original game based on the famous Labyrinth app for smartphones, as shown in figure 18.

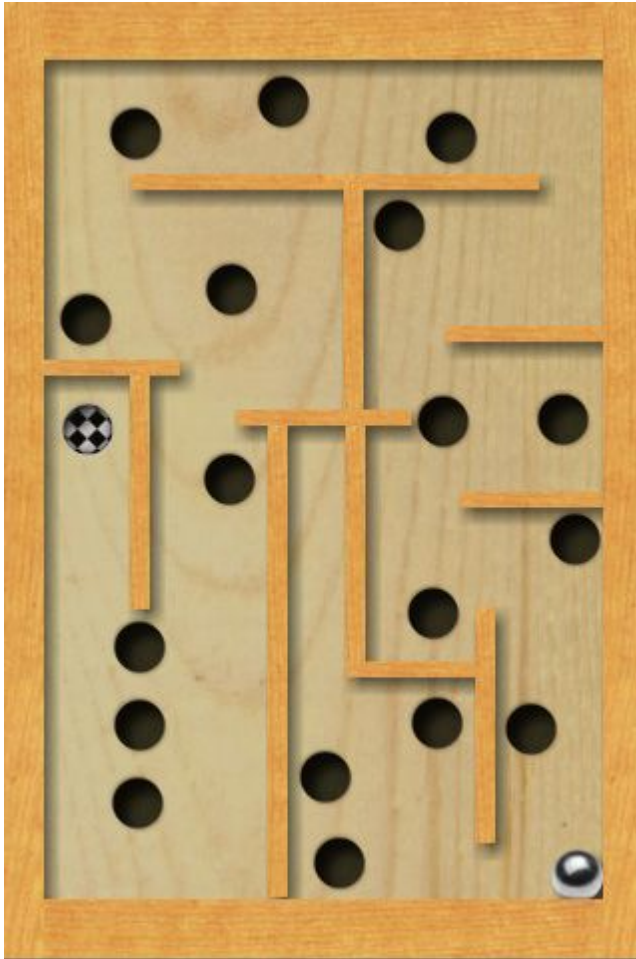


Figure 18. Scene from the game Labyrinth, for iOS

This application works such that tilting the device causes the ball to move through a maze created in the screen of the smartphone to avoid holes and reach the goal. The tilting simulates an effect of gravity on the ball, thus allowing it to move through the stage.

From this application it has been obtained the concept of controlling a ball through gravity, and this is why it was decided to create a game where, by tilting the field in the real world, you can move a ball in the virtual world to avoid that it is crushed by boxes falling from above the field.

The difference with the game Labyrinth, apart from the objective of the game, is that the device where the game is shown is not the same one that must be tilted to control the game. In this case it is not necessary the use of another device to control the game but a distinctive image, or tracker, to act as the game field.

Game concepts

After deciding how the application was going to be implemented, the only remaining thing was to decide the theme and other graphical concepts of the games.

The first choice was to choose the image that would act as a tracker so that the game could take place on it. As its recognition is completely managed by the library Vuforia, their advice on the provisional tracker was taken [18]: a picture rich in details, with good contrast and no repetitive patterns.

Following this previous advice, in the end it was decided that one of the trackers purposed in the gallery of Vuforia would be used, to optimize as much as possible the process of image recognition. As it can be seen in figure 24, this image follows every advice previously specified.



Figure 24. Tracker chosen to be used with Vuforia

With this design in mind, the next step was to choose how the main menu would look like. In this case, the options had to appear in a clear way to the user by taking as little space of the real world as possible, and that is why it was decided for the options to be placed over the image, like growing perpendicularly out of it (figure 25).

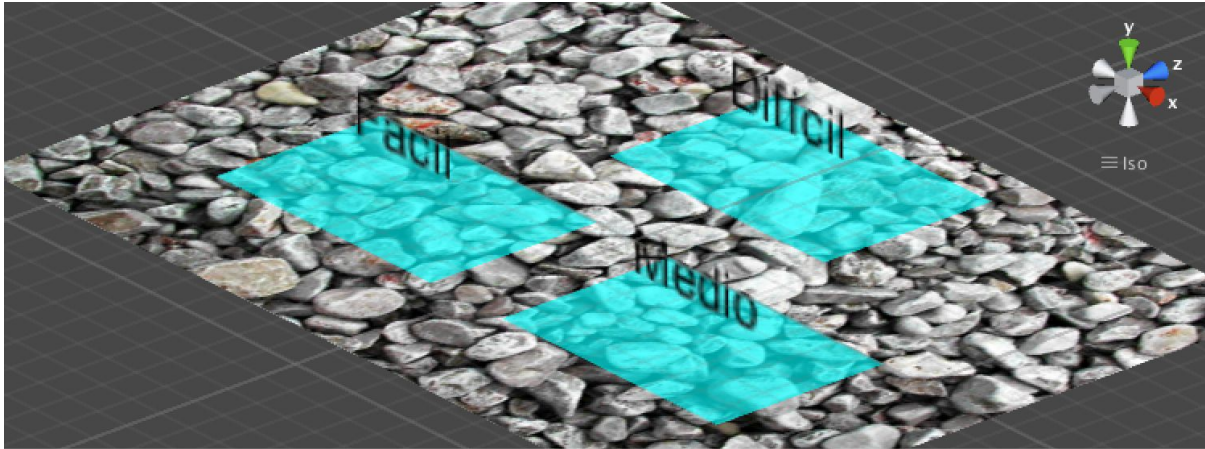


Figure 25. Initial positioning of the text with options

However, upon further reflexion it was concluded that most of the users would look down to the main menu with a 45° angle, given that this is the position that feels most natural. That is why the design was changed so that the words would make an angle of 45° with the tracker, as shown in figure 26, allowing it to be read both from an angle parallel to the marker and from a perpendicular one, as well as from every angle in between both.

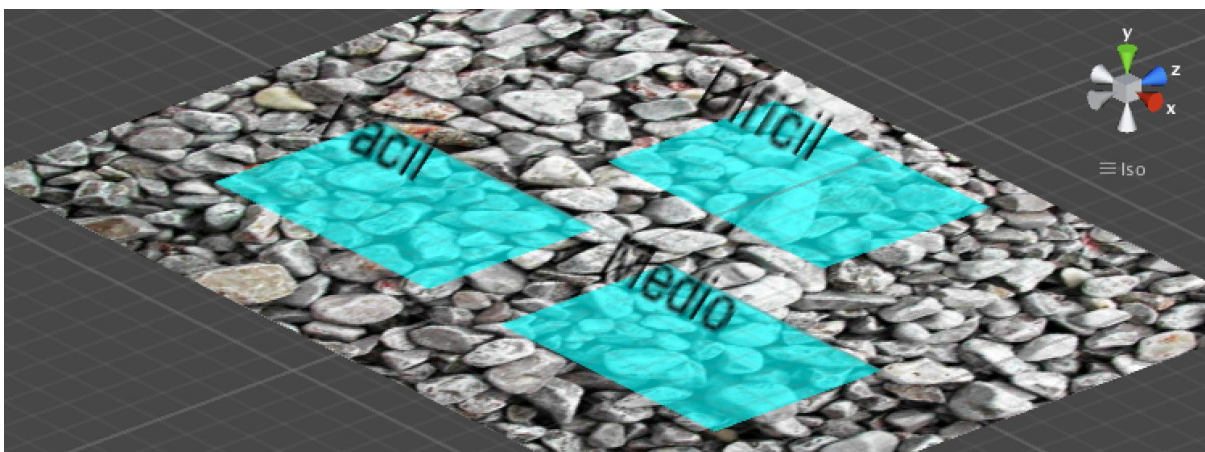


Figure 26. Final positioning of the text with options

As for the design of the game Arkanoid, it was made so that it was as close to the original game as possible. The bricks, depending on the number of hits remaining until they are destroyed, change their color, and the colors green (1 hit, easy), blue (2 hits, medium) and red (3 hits, hard) were chosen to represent how hard it is to make that brick completely disappear, based only on the number of hits.

Given that there is no hand recognition involved, but rather the occlusion of the tracker is checked to find its position, it was not possible to create a virtual field in the game. This is because if the hand itself was recognized, its area could be superimposed over the created virtual field.

But as this is not actively done, virtual elements always appear superimposed over real elements. When placing a field on the game, even though the hand would still be recognised to make the ball rebound, the user would not be able to see it

because it would be hidden behind the field, and control would be harder and less precise for him.

These reasons caused that the game field, as well as the walls surrounding the field (that would otherwise cut the hands of the user) were designed as invisible to the user, even though they still make the ball rebound and keep it in play.

In the last place, for the game of the maze the hand does not play such a relevant role in the control of the game, so walls and the field could be placed in the game so that the user could easily see the limits of the field.

As it is not a game that existed before, several options for the game theme were considered. Given the aesthetic simplicity and the number of free textures available for it, the chosen theme for the game was the American West. Once this was chosen, the boxes were covered with a texture to make them look like wooden crates, the field was modified so that it looked like a sand desert, the walls were designed as wooden fences and the ball became a tumbleweed, as shown in figure 27. Because of this, the game took the name Tumbleweed.

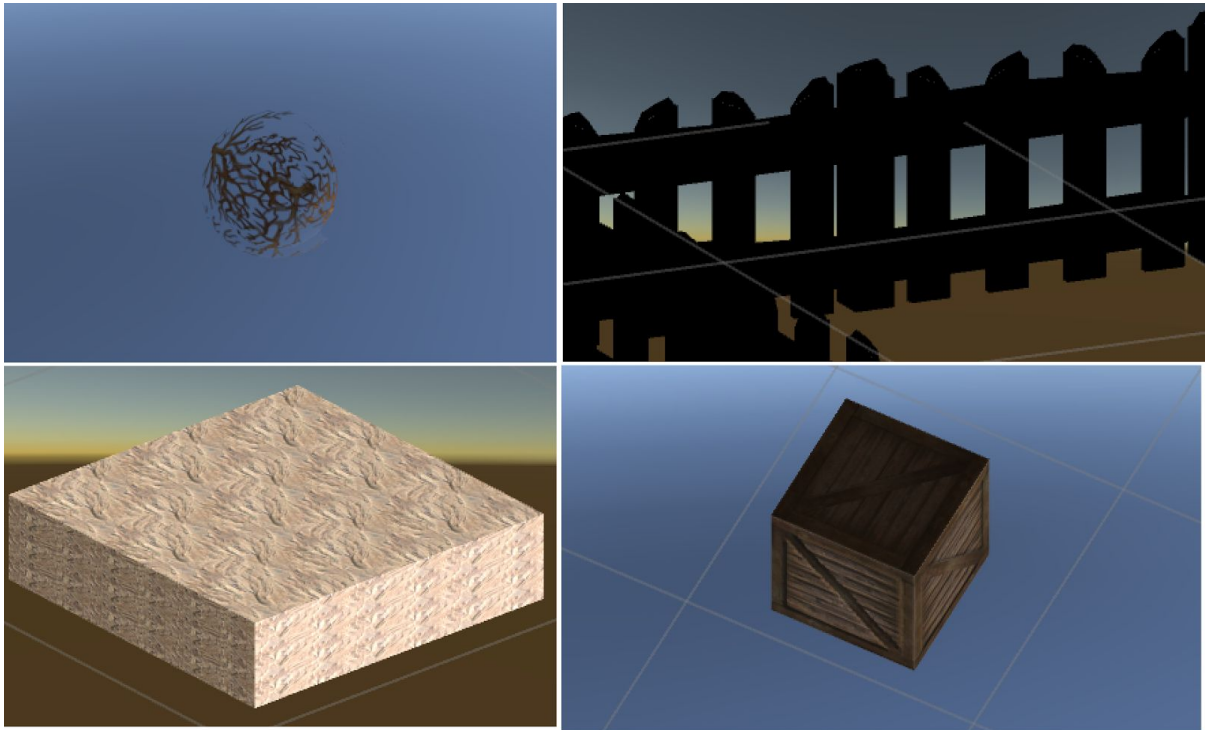


Figure 27. Design of the ball (upper-left), walls (upper-right), floor (lower-left) and boxes (lower-right) of the game Tumbleweed

Tests

Once the development of the application was completed, some tests were run to check that every functionality was correctly implemented for all the games.

First of all, the three difficulties of both games were tested. As can be seen in figures 41 and 42, by choosing different difficulties some changes in the initial setup of the field can be seen, and other changes like the speed of the ball or the boxes that cannot be appreciated in the pictures also took place.

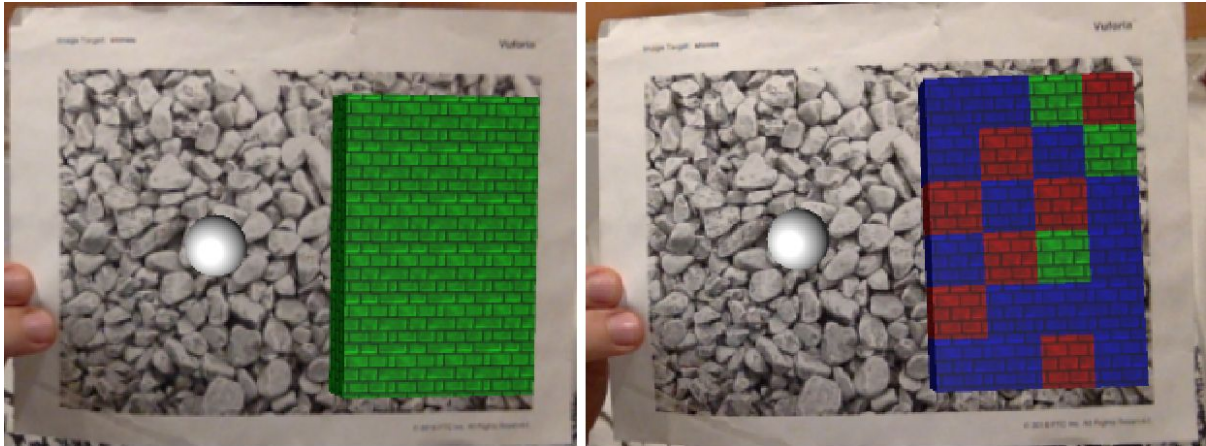


Figure 41. Game Arkanoid in easy mode (left) and hard mode (right)



Figure 42. Game Tumbleweed in easy mode (left) and hard mode (right)

Afterwards, it was tested that every option of the main menu were easily visible and that they all redirected to the corresponding scene. It was also observed if the texts were visible from every possible front angle, as shown in figure 43.

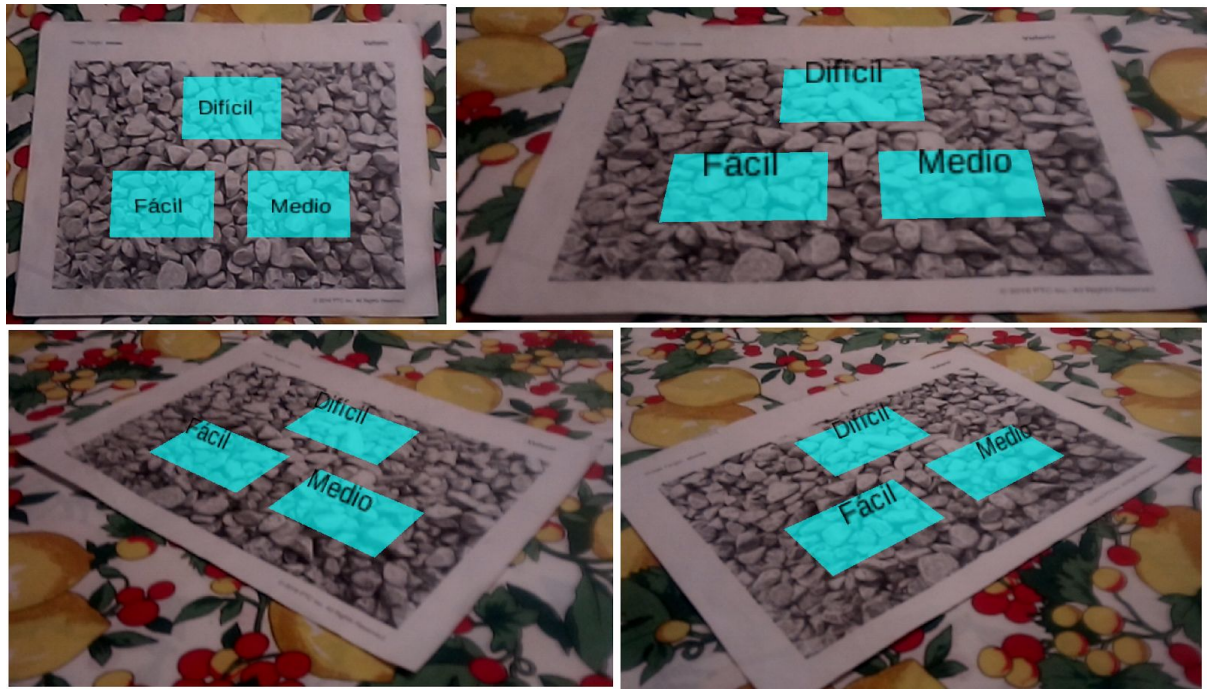


Figure 43. Difficulty selection screen from different angles

Another important aspect from the games was that they could be paused with just looking away from the field. After checking that this worked in both games, it could be seen that both continued exactly like they were left before looking away from their respective fields.

Lastly, it was checked that when the conditions that decide a victory or a defeat in a game were fulfilled, the games correctly indicated with text the end of the game (figure 44).



Figure 44. Screenshot of victory (left) and defeat (right) of the game Arkanoid

Conclusions

Once the project has been successfully finished, the initial objectives that had been described before can be checked to see if they have all been fulfilled:

The main objective was to create a game of augmented reality that could be controlled completely without depending on virtual elements, through real world movements instead of touching the screen. This objective has been completely fulfilled, given that the selection menu is controlled by hovering the hand over the tracker, just like for the game Arkanoid. On the other hand, the game Tumbleweed is controlled by just tilting the tracker, without interacting with the device either.

The second objective was that the application interfered as little as possible with other activities that the user could be completing or that could just come up. This has been achieved by two different means: the first one is that the virtual elements created occupy as little as possible in the real world to interfere as little as possible with the vision of the user.

The second way to achieve this has been that, even though a tracker is needed to use the application, in case the user needs to focus on another thing or look away for whatever reason, as soon as the tracker disappears from the field of view of the user the app will continue in exactly the same state as left when the user looks at the tracker again, so that the user does not need to pay attention to the tracker at all times.

The last objective was to develop the application for both augmented reality glasses and smartphones, which has been fulfilled. The main development, as well as the test, were made in a smartphone from which the project was ported to an augmented reality glasses once it was finished, changing elements such as the angle in which the image is seen (because you need two images from different angles in augmented reality glasses to achieve the three-dimensional effect).

As it can be seen, the three main objectives have been successfully fulfilled. Although the initial planning of the project was of 4 and a half months, it has taken several extra months to complete it. This also indicates the difficulty in planning a software development project, since the estimated planning did not include incidents that have delayed the final product for months.